



ShakeCast V3 Operator's Technical Guide

By Kuo-Wan Lin, David Wald, Vikki Appel and Loren Turner

Version 1.0, February 2015

U.S. Department of the Interior
U.S. Geological Survey

U.S. Department of the Interior
SALLY JEWELL, Secretary

U.S. Geological Survey
Suzette M. Kimball, Acting Director

U.S. Geological Survey, Reston, Virginia: 2015

For more information on the USGS—the Federal source for science about the Earth, its natural and living resources, natural hazards, and the environment—visit <http://www.usgs.gov> or call 1-888-ASK-USGS (1-888-275-8747)

For an overview of USGS information products, including maps, imagery, and publications, visit <http://www.usgs.gov/pubprod>

To order this and other USGS information products, visit <http://store.usgs.gov>

Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Although this information product, for the most part, is in the public domain, it also may contain copyrighted materials as noted in the text. Permission to reproduce copyrighted items must be secured from the copyright owner.

Suggested citation:

Lin, Kuo-Wan, Wald, David, Appel, Vikki and Turner, Loren 2015, ShakeCast V3 Operator's Technical Guide (ver. 1.0 February, 2015):

U.S. Geological Survey report series and number, total number of pages, <http://dx.doi.org/10.3133/XXXXX>.

ISSN XXXX-XXXX (print) [if applicable]
ISSN XXXX-XXXX (CD/DVD) [if applicable]
ISSN XXXX-XXXX (online)
ISBN XXX-X-XXXX-XXXX-X [if applicable]

Contents

1	ShakeCast System Installation.....	1
1.1	ShakeCast on a Virtual Machine	1
1.2	ShakeCast System Hardware Requirements	1
1.3	ShakeCast System Software Requirements.....	2
1.4	Security and Firewall	3
1.5	Web Browser Compatibility.....	3
2	ShakeCast MS-Windows Installation.....	4
2.1	Prerequisites Wizard.....	4
2.2	ActivePerl.....	5
2.3	Apache.....	7
2.4	MySQL.....	9
2.5	PHP	14
2.6	ShakeCast	18
2.7	ShakeCast Cloud Installation.....	20
2.7.1	ShakeCast Amazon Web Services (AWS) Cloud Installation Steps.....	20
3	Post-installation System Configuration.....	23
3.1	Activate Earthquake Processing.....	23
3.2	Configuring Email Server for Notifications	24
3.3	Restarting System Services.....	26
3.4	Database Management	26
3.5	System and Database Reset	27
3.5.1	System Reset.....	27
3.5.2	Database Reset	27
3.6	Facility Inventory.....	28
3.7	User Groups	30
3.8	User Inventory	33
3.9	Earthquake and Scenario Inventory	33
3.10	Notification and System Records.....	36
4	ShakeCast End User Customization	38
4.1	ShakeCast Email Notifications.....	38
4.2	Default Facility Assessment Message	39
4.3	Notification Template	39
4.3.1	Template Directory.....	40
4.3.2	Customizing Email Notifications.....	40
4.3.3	Template Layout and Content.....	42
4.4	ShakeCast PDF Summary Report.....	42
4.4.1	PDF Summary Report Template.....	43
4.4.2	Template Layout and Content.....	44

5	ShakeCast System Administration	46
5.1	Scope of System Management.....	46
5.2	General Settings	47
5.2.1	System Services Restart.....	47
5.2.2	ShakeCast Database	48
5.2.3	Configuring the Notification Email Server	48
5.2.4	ShakeMap Server	50
5.2.5	ShakeMap Region.....	51
5.2.6	System Directory and Miscellaneous Parameters	52
5.3	Earthquake Database Management	53
5.4	Facility Database Management	56
5.5	Earthquake Product Management	60
5.6	Station Database Management	61
5.7	User Database Management.....	62
5.8	Inventory Upload Utility	65
5.9	Cron Job	67
6	References	69
	Appendix A ShakeCast User Data Formats.....	70
	Appendix B ShakeCast XML/JSON Metadata Formats.....	90
	Appendix C ShakeCast Script Utilities	114
	Appendix D Relating HAZUS Structure Types to Seismic Performance	145
	Appendix E HAZUS Model Building Fragility Table	149
	Appendix F ShakeCast Notification Template Keywords.....	155

ShakeCast V3 Operator's Technical Guide

This guide is intended for ShakeCast system managers, operators and IT support staff. This guide is for you if you are *installing*, *operating*, or *configuring* a ShakeCast system.

If you are a ShakeCast user interested in learning more about ShakeCast notifications, facility assessment or inspection reports or web pages, see the **ShakeCast User Guide**
<https://my.usgs.gov/confluence/display/ShakeCast/>.

1 ShakeCast System Installation

The ShakeCast V3 system was developed under the CentOS 6 Linux system and ported to other Linux distributions (RedHat ES 6 and SUSE Enterprise Server 10) and the Windows operating system (7/Server 2008) 64-bit application.

Supporting multiple OS platforms is not possible due to resource limitations, so for ShakeCast users, the strategy going forward is to support Linux Virtual Machine (VM) for local installations and Cloud computing on Amazon Web Services (AWS). The ShakeCast application is bundled with an Open-Source Linux operating system as a standalone system image with the standard installation package.

1.1 ShakeCast on a Virtual Machine

A virtual machine (VM) is a software implementation that executes programs like a physical machine. Virtual machines are separated into two major classifications, system and process virtual machines, based on their use and degree of correspondence to any real machine. ShakeCast VM is a system virtual machine.

ShakeCast V3 on a VM has the benefits of application provisioning, maintenance, high availability and disaster recovery. These also are important factors for consideration when implementing the application at any organization as a VM or physical server. The host VM described in this document reflects one possible VM option for the purpose of application development. USGS does not endorse any specific VM host for the ShakeCast application. The sections below describe the setup for a generic VM available to the ShakeCast user community.

1.2 ShakeCast System Hardware Requirements

Recommended minimum hardware specifications for the ShakeCast system includes:

- Single Intel Xeon E5-2670 equivalent processor.
- 1GB RAM.
- 30GB hard drive storage
- At least low performance Internet connection (<1MB/s)

The above hardware is roughly equivalent to the “micro” instance on the Amazon Elastic Compute Cloud (Amazon EC2) in which performance was assessed. A system with these minimum requirements

could be used for non-production purposes and was found to be adequate to support a ShakeCast instance with <100 facilities and <10 users for each processed ShakeMap.

Depending on the number of facilities, user inventory and the earthquake monitoring areas, more hardware resources will be needed to for better performance i.e., near instantaneous facility evaluation and user notifications. Products (ShakeMap, ShakeCast, PAGER, DYFI?, and others) for each processed earthquake usually consume 30-50 MB of hard drive space. For ShakeCast systems designated for earthquake response purpose, we recommend to at least double the minimum recommended hardware specifications. As a case example, the Caltrans ShakeCast system consists of ~26,000 bridge facilities, ~500,000 bridge components, ~300 users in several groups and uses the following hardware for all primary and backup servers:

- Quad Intel Xeon X5670 2.9GHz processors.
- 8GB RAM.
- 100GB hard drive storage.
- High performance Internet connection.

1.3 ShakeCast System Software Requirements

The ShakeCast V3 system is distributed for both Linux and MS-Windows operating systems. The system is built on an open-source stack of supporting applications shared by all platforms, specifically:

- Apache Web server 2.x.
- MySQL 5.x database.
- Perl 5.14+ scripting language.
- Perl Modules: DBI, DBD::mysql, Text::CSV_XS, Config::General, enum, XML::Parser, XML::LibXML, XML::Writer, XML::Twig, XML::Simple, Template-toolkit, PDF::API2, PDF::Table, MIME::Lite, GD, GD::Text, GD::Graph, GD::Graph3d, HTML::TableExtract, Net::SSLeay, Net::SMTP::SSL, Net::SMTP::TLS, Authen::SASL, Archive::Zip, JSON, JSON::XS, File::Path, Image::Size, Mojolicious.
- wkhtmltoimage conversion tool.
- gnuplot image tool.
- HTML5/Google Maps API V3/markerclusterer/jQuery/Bootstrap/dataTables Web tools.
- Optional PHP/phpmyadmin scripting language.
- Optional git version control tool.

Linux implementations:

- Xvfb X virtual framebuffer display server (required for 64-bit systems and optional for 32-bit systems).
- mailx as default mail utility.
- ShakeCast services as background daemon processes.
- Database backup cron job.

Windows implementations:

- SMTP as default mail protocol (supports both SSL/TLS security layers).
- ShakeCast services as Windows system processes.

1.4 Security and Firewall

The default setup of ShakeCast allows access via the command line using SSH and the web interface with HTTP or HTTPS. The ShakeCast web server is designed to serve earthquake information to users and to allow administrators to conduct general administration of the system.

Command line access via SSH (Linux) should be granted only to system administrators. ShakeCast tasks not covered by the web interface are considered advanced topics for experienced ShakeCast administrators. For Windows operating systems without installed SSH service, the ShakeCast administrator will need to access the system via the default **Remote Desktop Connection** application (or similar remote access programs) to perform the same tasks.

Normal setup and interaction with a user's ShakeCast web server provides user access to maps, products and services, as well as administrator access. Administrators can modify user profiles and notifications, trigger earthquake scenarios, and access many other configurations functions. However, in the most secure installation of ShakeCast, the administrator can choose to disable modifications from the web and only permit SSH access.

Firewall and system level security configurations are platform specific and not covered by this manual. ShakeCast implements basic authentication, but it is highly recommended to implement system-level firewall policies to limit exposure to/from the Internet. These rules will take precedence over the ShakeCast-defined user authentication. For inbound traffic, firewall policies are effective methods to define domains where users can access the products and information of the ShakeCast server. For outbound traffic, firewall policies should permit the USGS Web server (<http://earthquake.usgs.gov>), which is the source for all earthquake products processed by ShakeCast. For ShakeCast systems receiving earthquake products via the USGS Product Distribution Layer (PDL) client, the program uses port 39977 to connect to the upstream hub server.

1.5 Web Browser Compatibility

The ShakeCast V3 web interface was built using HTML5 standards. Most user and administrator interactions are through using a web browser. Supported web browsers:

	MacOS	MS-Windows
Chrome	25+	25+
Firefox	20+	15+
Opera	12+	12+
Safari	5+	N/A
Internet Explorer	N/A	9+

2 ShakeCast MS-Windows Installation

A Windows Installer Package of ShakeCast V3 for Windows allows the administrator download, install, and configure the ShakeCast system. The package requires administrative privileges on the system and the user performing the installation must be a member of the local Administration security group.

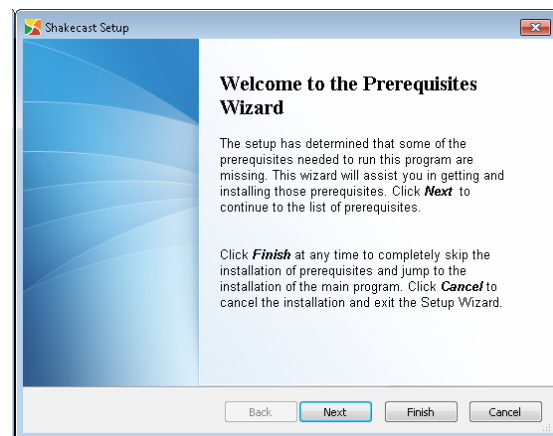
The installer prompts the users a series of custom setup options and in general default options are the best choices for new users. The size of a new ShakeCast installation is about 2GB including third-party programs that are used by the ShakeCast application.

2.1 Prerequisites Wizard

Double-click the ShakeCast installer. If a security-warning message appears, click **Run**.

The ShakeCast Prerequisites Wizard window opens.

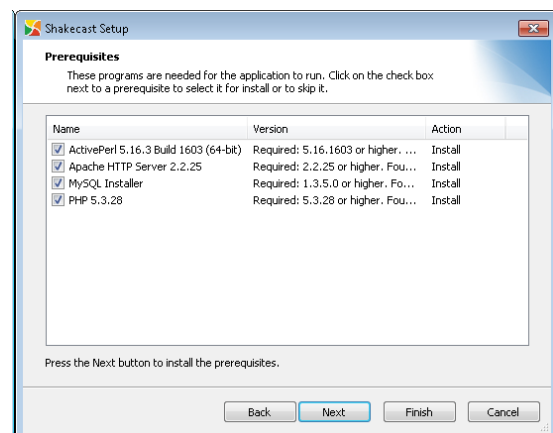
Click **Next**.



The Wizard analyzes pre-installed programs on the operating system. Existing programs that meet the software requirements will be unchecked.

Toggle checkboxes to make optional changes to the planned program list for installation.

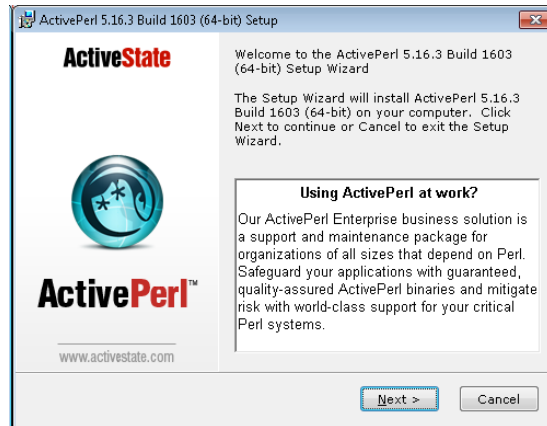
Click **Next**.



2.2 ActivePerl

The ActivePerl Setup Window opens.

Click **Next**.



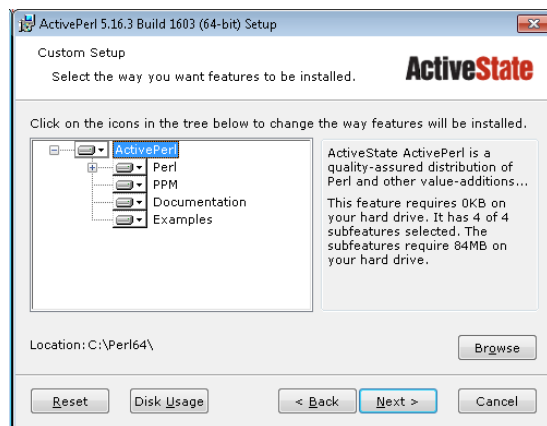
After reading the license, click the option to accept the terms of the license agreement.

Click **Next**.



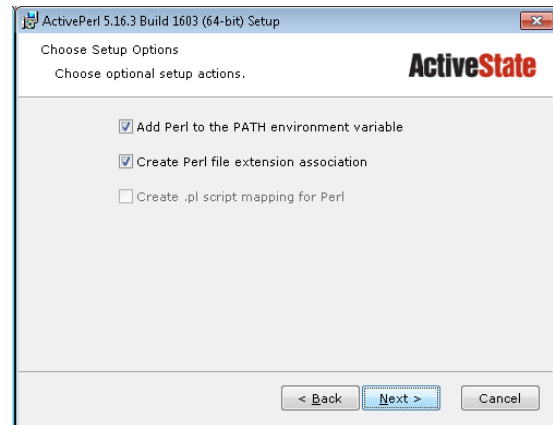
The **Custom Setup** screen appears.

Click **Next**.



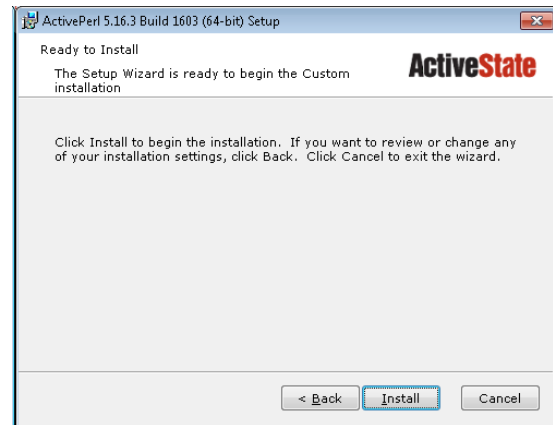
The **Choose Setup Options** screen appears. Do not uncheck the first two checkboxes.

Click **Next**.



The **Ready to Install** screen appears.

Click **Install**.



The next window shows the Setup Wizard is complete.

Click **Finish**.

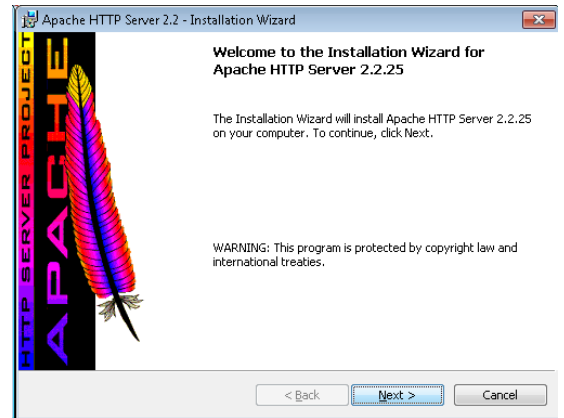
The **ActivePerl Release Notes** are displayed.



2.3 Apache

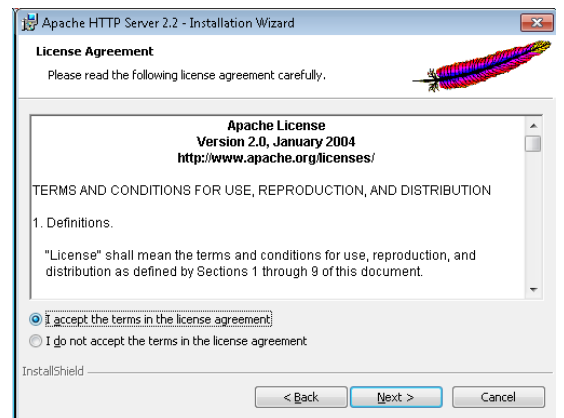
The **Apache HTTP Server Installation Wizard** appears.

Click **Next**.



The Apache License Agreement screen appears. Click to accept the terms in the license agreement option to accept the terms.

Click **Next**.



The Apache Readme document screen appears.

Click **Next**.



The Apache Server Information screen appears.

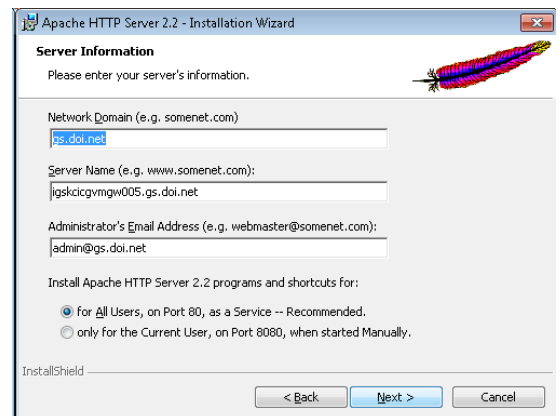
Type your network domain name.

Type your server name.

Type administrator's e-mail address.

Select the for All Users option to install for all users.

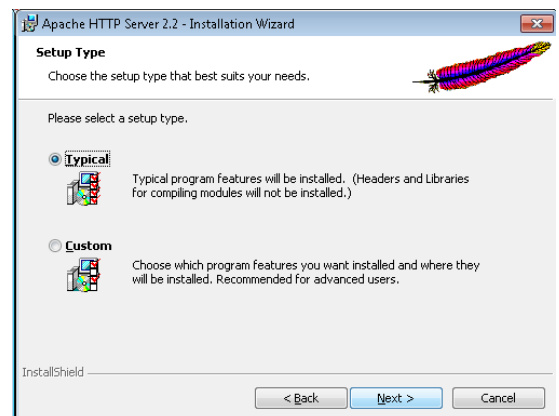
Click **Next**.



The Apache Setup Type screen appears.

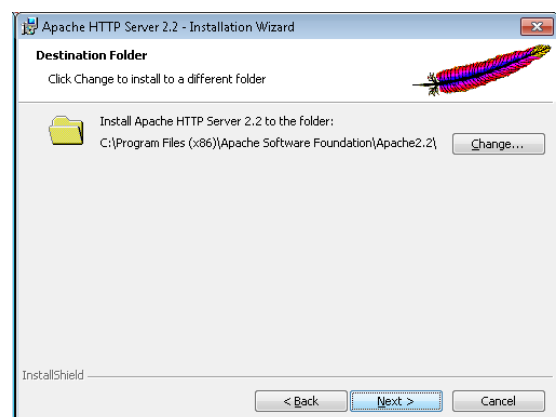
Click to select the Typical option.

Click **Next**.



The **Apache Destination Folder** screen appears.

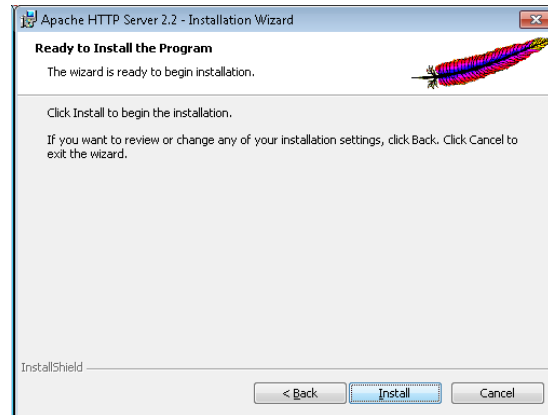
Click **Next**.



The wizard is ready to begin the installation.

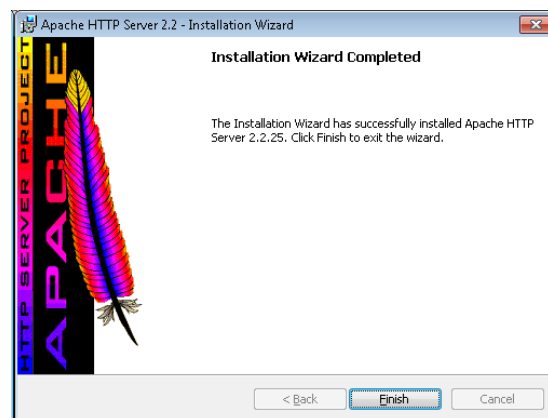
Click **Install**.

Apache for Windows is installed.



The Apache for Windows is installed.

Click **Finish**.



2.4 MySQL

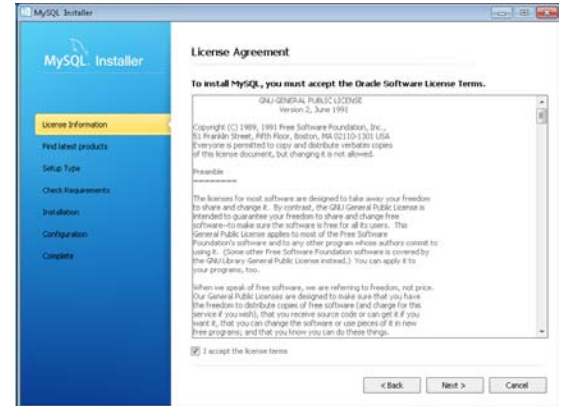
The MySQL Setup Wizard appears.

Click **Install MySQL Products**.

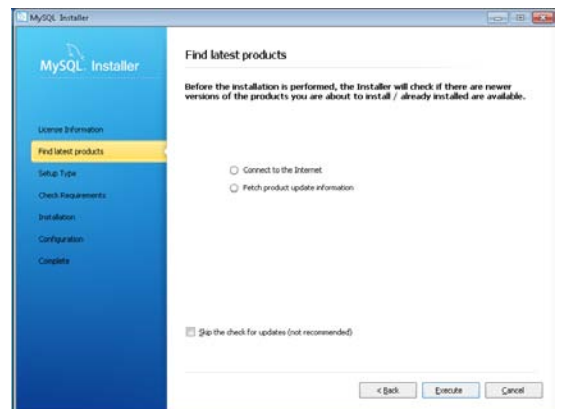


The MySQL License Agreement screen appears. Click to accept the terms in the license agreement option to accept the terms.

Click **Next**.



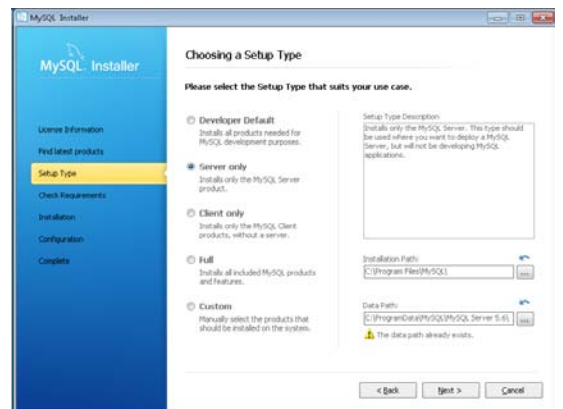
The MySQL Find latest products screen appears. Click **Execute**.



The MySQL Choosing a Setup screen appears.

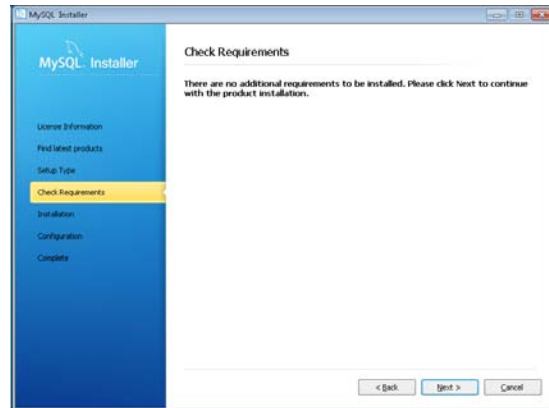
Select the **Server only** option.

Click **Next**.



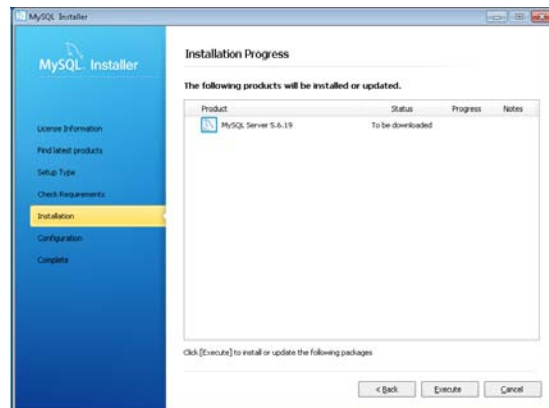
The MySQL Check Requirements screen appears.

Click **Next**.



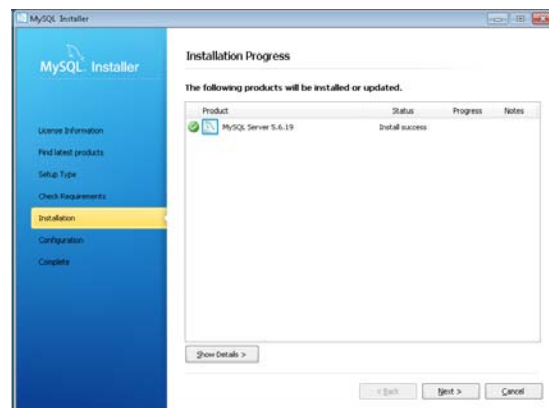
The MySQL Installation screen appears.

Click **Execute**.



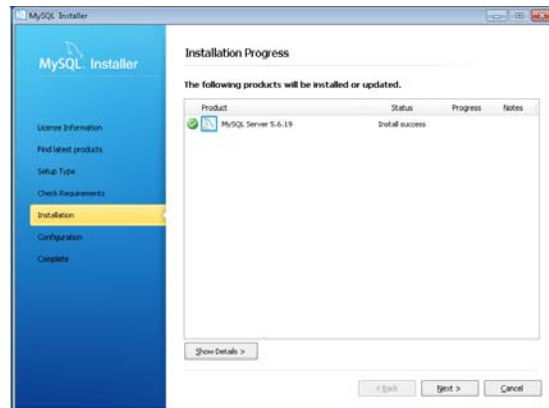
The MySQL Installation Progress is completed.

Click **Next**.



The MySQL Configuration screen appears.

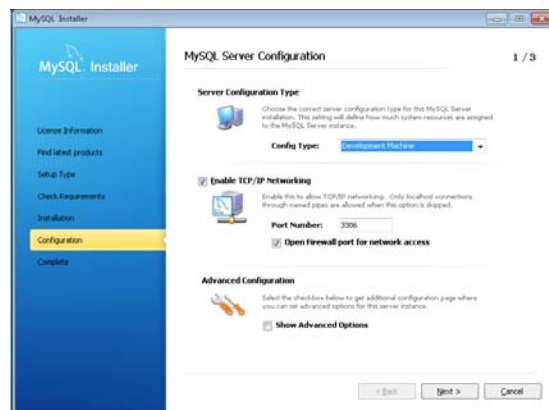
Click **Next**.



The MySQL Server Configuration screen appears.

Select Server Machine as the Config Type.

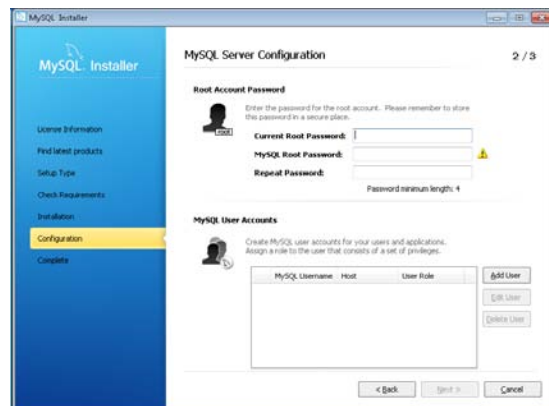
Click **Next**.



Step 2 of the MySQL Server Configuration screen appears.

Type in a root password.

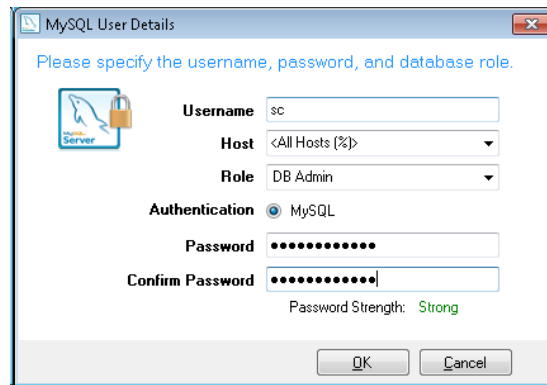
Click **Add User**.



Type in “sc” in Username field.

Type in a Password.

Click **OK**.



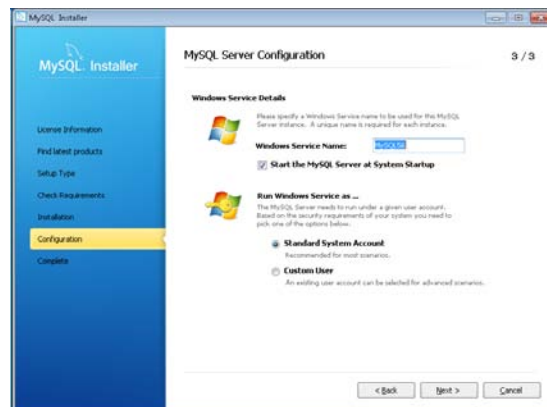
Return to Step 2 of the MySQL Server Configuration screen.

Click **Next**.



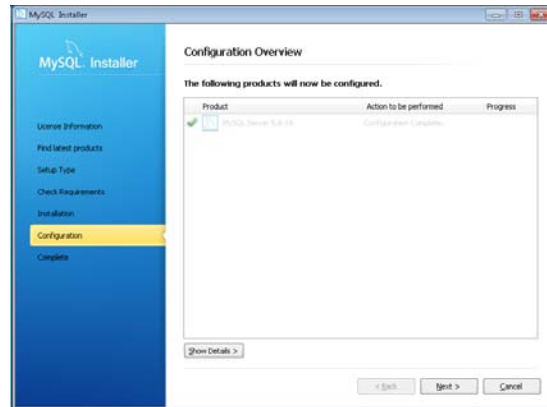
Step 3 of the MySQL Server Configuration screen appears.

Click **Next**.



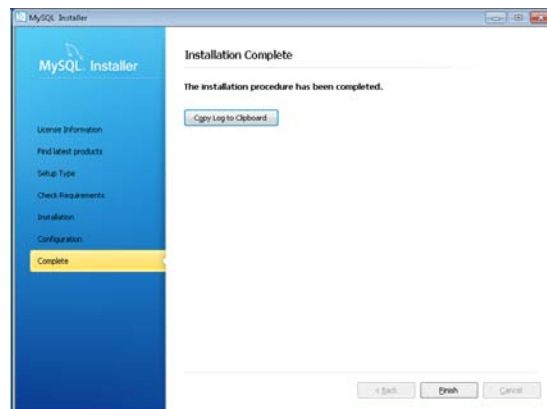
The MySQL Configuration Overview screen appears.

Click **Next**.



The MySQL Installation Complete screen appears.

Click **Finish**.



2.5 PHP

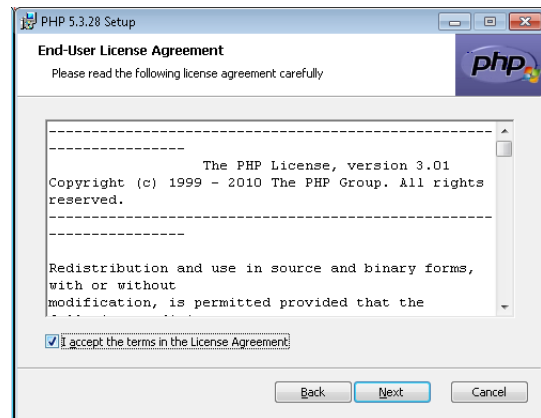
The PHP Setup Wizard is displayed.

Click **Next**.



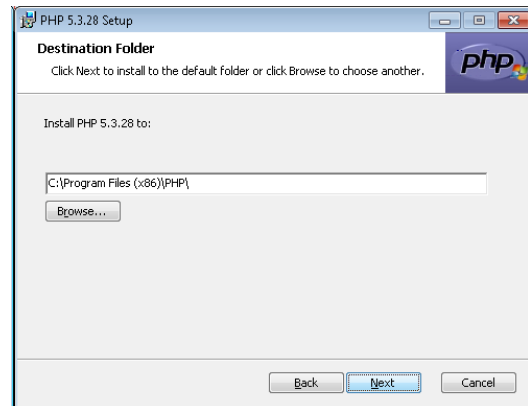
The PHP End-User License Agreement screen appears. Click to accept the terms in the license agreement option to accept the terms.

Click **Next**.



The PHP Destination Folder screen appears.

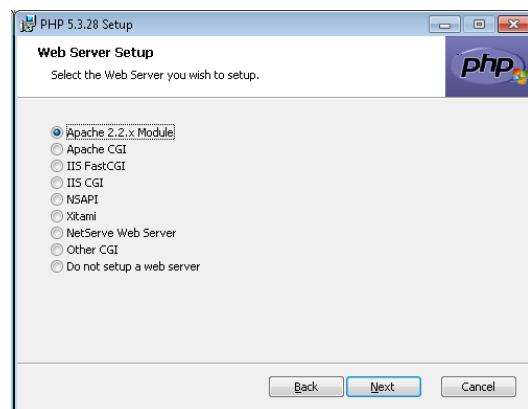
Click **Next**.



The PHP Web Server screen appears.

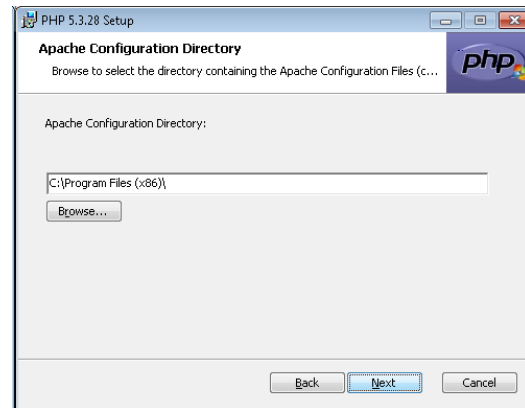
Select the Apache 2.2.x Module.

Click **Next**.



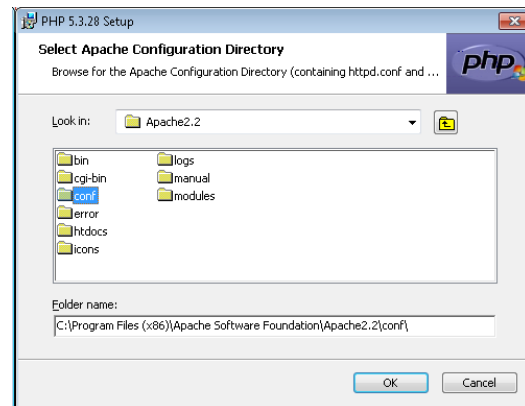
The PHP Apache Configuration Directory screen appears.

Click **Browse** to select the Apache Configuration Directory.



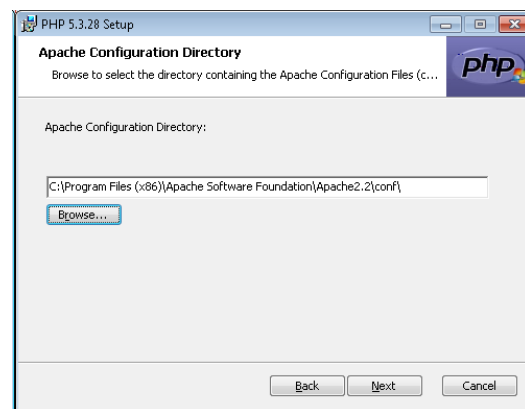
Navigate to the “C:\Program Files (x86)\Apache Software Foundation\Apache2.2\conf” directory.

Click **OK**.



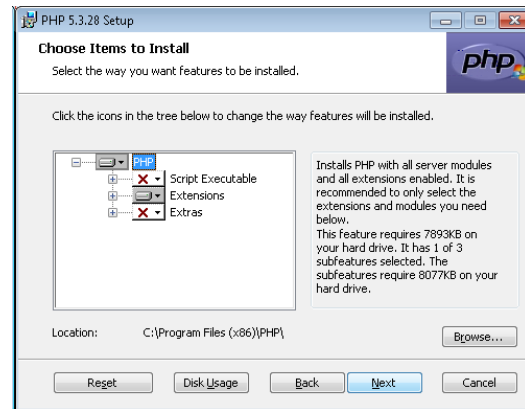
Return to the PHP Apache Configuration Directory screen.

Click **Next**.



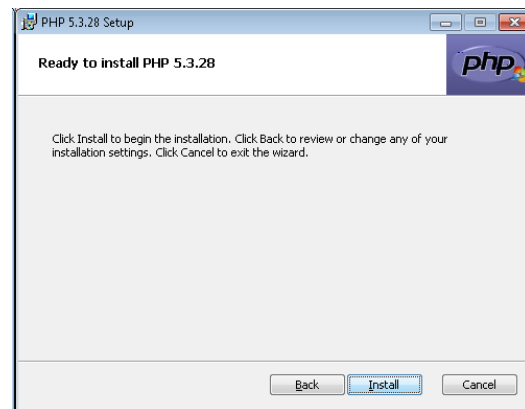
The PHP Choose Items to Install screen appears.

Click **Next**.



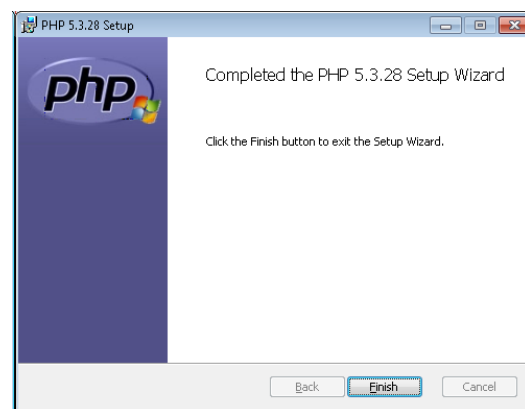
The Ready to Install PHP screen appears.

Click **Install**.



The PHP installation is completed.

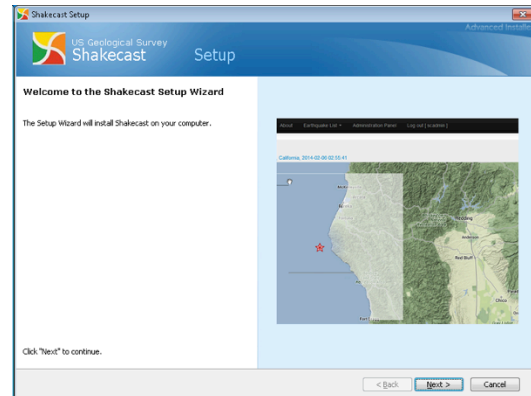
Click **Finish**.



2.6 ShakeCast

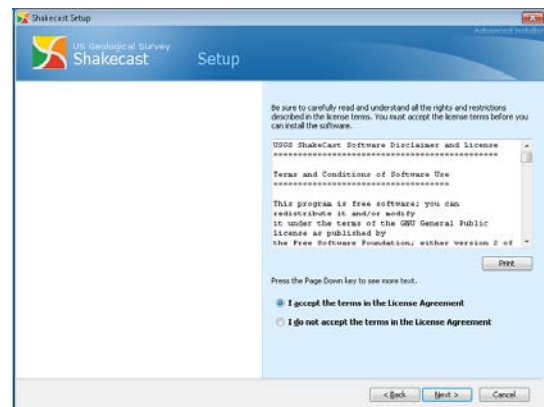
The **ShakeCast Setup Wizard** is displayed.

Click **Next**.



The ShakeCast End-User License Agreement screen appears. Click to accept the terms in the license agreement option to accept the terms.

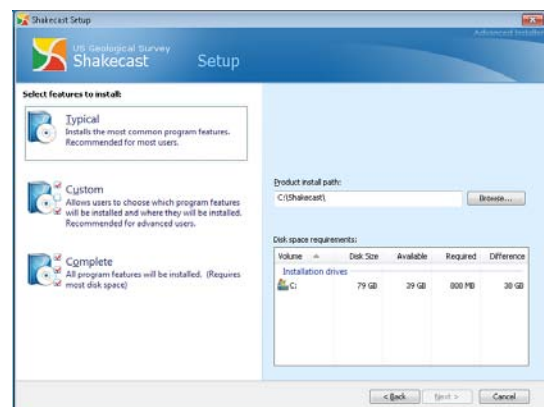
Click **Next**.



The ShakeCast Setup feature screen appears.

Click **Typical** to select the most common program features.

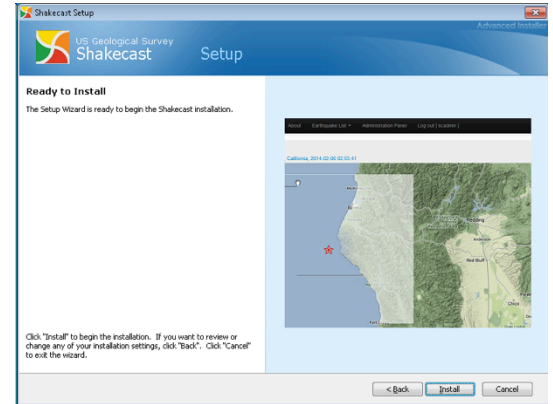
Click **Next**.



The ShakeCast Ready to Install screen appears.

Click **Install**.

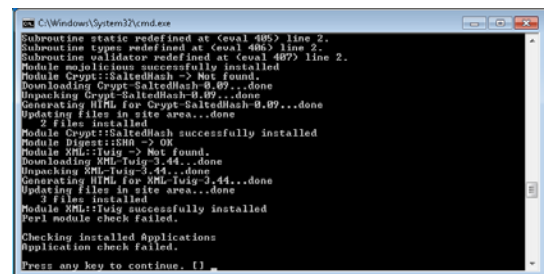
Installation of the base ShakeCast system may take several minutes to finish.



The ShakeCast Perl Post installation screen appears to start the automatic configuration process.

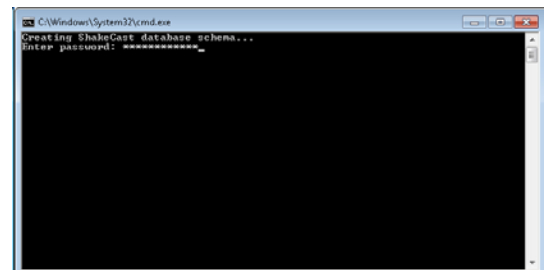
When the process is finished, the message “Press any key to continue. []” at the end of the screen.

Click any key to continue.



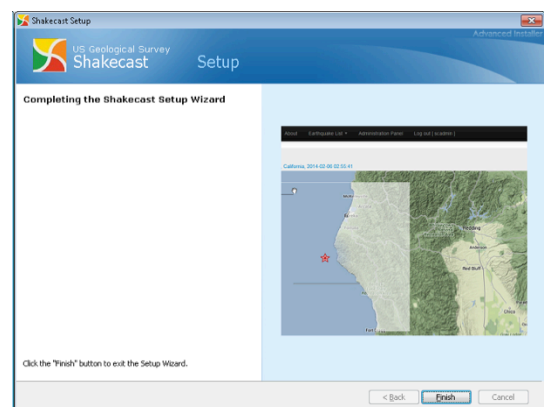
The ShakeCast Database Post installation screen appears.

Enter the root account password for the MySQL database and hit **Enter** to Continue.



The Completing the ShakeCast Setup Wizard screen appears.

Click **Finish**.



2.7 ShakeCast Cloud Installation

ShakeCast V3 is shipped primarily as a pre-packaged Linux system image. Among tested distributions, CentOS (open-source equivalent of the RedHat Enterprise Linux) is the default operating system (OS). With the ready-to-use ShakeCast V3 system image, users can evaluate the software without excessive IT commitment to determine the use case that will satisfy their needs for post-earthquake response.

Amazon Web Services (AWS) offers a one-year, free-tier elastic compute cloud product (EC2) suitable for the ShakeCast program. A custom built private ShakeCast system image using Amazon EC2 streamlines installation and system updates. Users can launch their own ShakeCast system as a “micro” instance on Amazon EC2 without incurring charges during the evaluation period of up to one year.

2.7.1 ShakeCast Amazon Web Services (AWS) Cloud Installation Steps

This documentation reflects the current AWS configuration. Updated and installation instructions reflecting any changes to the AWS or ShakeCast system are available as a standalone document at: <http://earthquake.usgs.gov/research/software/shakecast/>

Installing ShakeCast V3 on Amazon Web Services (AWS)

1. **Sign up for an Amazon Web Services (AWS) account**

You must have an AWS account to check out the ShakeCast system image. Sign up at <http://aws.amazon.com>

2. **Request Access to the ShakeCast private system image (AMI)**

Once you have your AWS account, send a request to shakecast-help@usgs.gov to access the ShakeCast private system image (AMI). Include the AWS account number in the email request, as you will be using this account to launch your own instance of the ShakeCast system.

3. **Access the ShakeCast AMI**

The ShakeCast AMI will appear under the EC2 Dashboard after you receive the confirmation email from the ShakeCast team.

To verify access to the ShakeCast AMI:

- Select **EC2** Dashboard from AWS Service Console.
- Select **AMIs** under the Images tab and use the filter **Private Images**. Contact ShakeCast team if the ShakeCast AMI is not shown in the image list.

Note: Make sure the Region in the top right is set to **N. Virginia** in order to see the private image

4. **Create ShakeCast Security Group**

Begin the ShakeCast installation process by creating a custom security policy permitting only ShakeCast traffic over the Internet, SSH, HTTP, and HTTPS. Select **Security Groups** link under Network Security left-hand navigation list.

Click the Create Security Group button to create the ShakeCast Security Group.

This step should be evaluated by an IT Administrator before using for a production system. For development purposes, allowing SSH, HTTP, and HTTPS will suffice. For development purposes create a custom security policy permitting only ShakeCast-specific traffic over the Internet, SSH, HTTP, and HTTPS. Select **Create a new security group**.

Select the newly created ShakeCast Security Group. Click the Inbound tab to add security rules. This will allow customization of the security groups

5. Create SSH Key Pairs

SSH key pairs are used to remotely log into the ShakeCast system to gain system-level access to perform tasks not available from the web interface. Tasks include both ShakeCast and system related work and access to the system is likely required at some point during the operation. Since the ShakeCast system image does not include a graphical user interface for system level access, operations can only be performed from the command line and are reserved primarily for IT administrators and expert ShakeCast users.

Select **Key Pairs** under the Network Security left-hand navigation list.

Click the **Create Key Pair** button to create an SSH key pair for accessing the ShakeCast system from the command line.

Download and save the private key for later use. The created key pair will be displayed in the key pair list window.

6. Launch the ShakeCast Instance

- Select the ShakeCast AMI from the AMI image window and click the Launch button to launch a new ShakeCast instance.
- Select **T1 Micro** as the Instance Type. Click the Continue button. Fill in instance details and tags for user-related information if applicable. Free tier eligible customers will get a maximum amount of 30GB of storage. Select **Next: Tag Instance**.
- Fill in instance details and tags for user-related information if applicable. Select **Next: Configure Security Group**. An IT Administrator should evaluate this step before using in a production system. For development purposes, allowing SSH, HTTP, and HTTPS will suffice. For development purposes create a custom security policy permitting only ShakeCast-specific traffic over the Internet, SSH, HTTP, and HTTPS. Select **Create a new security group**.
- Select the security group policy created earlier as the firewall setup for your ShakeCast instance.
- Review the setup of your instance. Click the **Launch** button to launch your ShakeCast instance. To abort the launch at any step during setup, select the **Cancel** checkbox. Repeat Step 8-14 to launch another new instance.

- Select the key pair created earlier for accessing your ShakeCast instance. Select the checkbox for acknowledgement for access to the private key file. If the launch fails it is due to not accepting the terms. Copy and paste the URL in a browser.
- Select **Continue**. Select **Accept Terms & Launch with 1-Click**. Close the screen with the green checkbox. You are now subscribed to the CentOS product.

Congratulations! You have successfully launched your first ShakeCast instance. For details on Amazon EC2, refer to AWS documentation at <https://aws.amazon.com/documentation/>

7. Access the ShakeCast Web Interface

- Select the ShakeCast instance from the **Instances** window. The public domain information of the selected instance will be displayed in the status window beneath the instances window.
- Copy the ShakeCast domain and open the URL using another browser window.
- Use the default username **scadmin** and password **scadmin** to log into the system. “scadmin” is the default administrator account for the ShakeCast system and has full access privileges.

NOTE: The secure layer is turned on by default, and you will need to switch to the https protocol and append /html to the end of the Public DNS name for browser access using the browser.

For example --> <https://ec2-#####.###/###/###compute-1.amazonaws.com/html/>

8. Access the ShakeCast System Using the Key Pair

Select the ShakeCast instance from Instances window. Select Connect from the Actions pull-down menu. Follow the instructions to use the SSH client and the pre-configured key pair to access the operating system.

3 Post-installation System Configuration

The ShakeCast AWS VM pre-built ShakeCast system comes with a minimal set of configurations with a default email server and worldwide monitoring for earthquake of magnitude 3.0 or greater. Unlike the AWS VM, a non-cloud installation of ShakeCast does not include information for an email server or a polygon of earthquake monitoring.

Until a monitoring region is defined after installation, the ShakeCast system will download earthquake data and ShakeMap products, but will not attempt to process them. No shaking estimates at facilities, local ShakeCast products or ShakeCast notifications can be produced until an earthquake-monitoring region is defined.

To customize the ShakeCast system, users must prepare inventory files outside the ShakeCast system and use the administrative web interface or command-line tools to update the database. The **ShakeCast V3 system does not allow direct editing of user or facility inventory in the interface or directly to the database**. To upload files, users must use the drag-and-drop upload in the administration area (**Upload** tab).

3.1 Activate Earthquake Processing

To activate earthquake processing, the administrator must define at least one ShakeCast group. The geometric coordinates of an enclosed polygon defines earthquake monitoring area. When multiple areas are defined, the union of the monitoring regions becomes the new monitoring area.

The configuration file (ca.conf) for the group “CA” in the example below defines an earthquake monitoring that covers the State of California and bordering regions with notifications for new earthquakes of magnitude 3.0 or greater (LIMIT_VALUE) within the region.

```
# Group Configuration file for CA
# $Id: ca.conf 221 2014-07-23 21:04:39Z klin $

<CA>
    POLY    43.000 -126.000    \
            39.000 -126.000    \
            34.000 -123.000    \
            31.000 -118.000    \
            31.000 -113.000    \
            36.000 -113.000    \
            39.000 -118.000    \
            43.000 -118.000    \
            43.000 -126.000

    <NOTIFICATION>
        NOTIFICATION_TYPE    NEW_EVENT
        DELIVERY_METHOD      EMAIL_HTML
        EVENT_TYPE            ALL
        AGGREGATE              1
        LIMIT_VALUE            3
    </NOTIFICATION>
</CA>
```

ShakeCast can process group definition configuration files via the upload tool from the web interface or from the command line with the **manage_group.pl** tool. Although not strictly enforced, it is recommended to postfix ShakeCast group definition file with a .conf file extension. After the group configuration file has been successfully processed, the ShakeCast group is displayed under the **User** tab of the administrative interface as shown below.

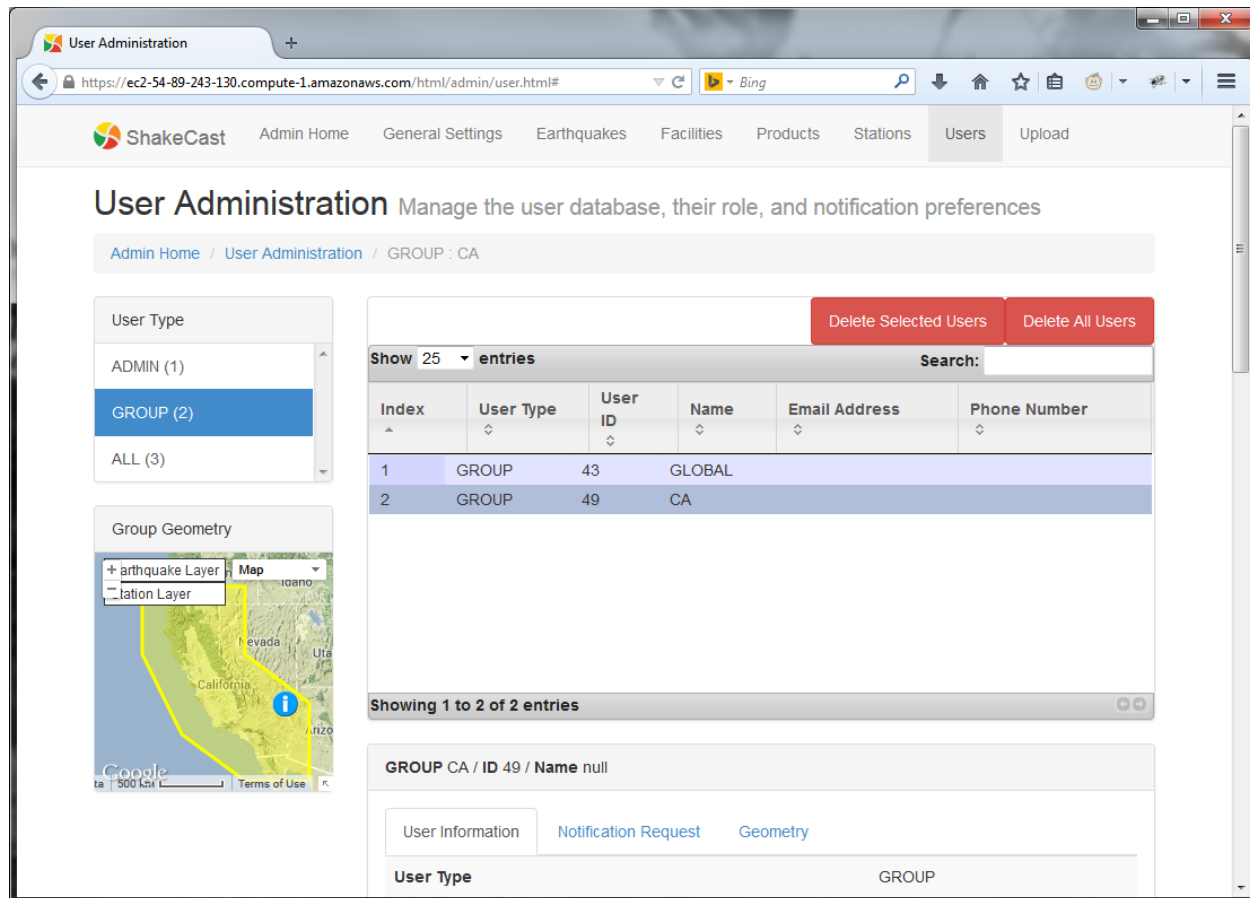


Figure 1. ShakeCast Group CA Example.

3.2 Configuring Email Server for Notifications

ShakeCast V3 supports the following email protocols:

- Mail Transport Agent (MTA) via mailx
- SMTP with plain text with and without authentication
- SMTP over Secure Sockets Layer (SSL)
- SMTP over Transport Layer Security (TLS)

By default, Linux-based ShakeCast installations (e.g., ShakeCast AWS VM) use the built-in mailx utility and require no additional manual email server configuration. However, the administrator can

choose to overwrite the default by specifying an SMTP email server. The Windows system typically lacks MTA support, so the administrator is advised to enter an SMTP server for delivering ShakeCast notifications.

The administrator can enter the SMTP information from either the web interface under the **General Settings** tab or by editing the ShakeCast system configuration file located at: “<sc_home>/sc/conf/sc.conf.”

Required SMTP email server information:

- DNS hostname of the SMTP server
- Security Layer (none/SSL/TLS)
- Service port (25/465/587)
- Username and Password

The screenshot shows the 'General Settings' tab selected in the ShakeCast web interface. The left sidebar contains a list of settings: 'ShakeCast Database', 'Email Server' (highlighted), 'ShakeMap Server', 'ShakeMap Region', 'System Directory', and 'Misc. Parameter'. The main content area is titled '2. Email Server' and includes the instruction 'Specify information for Email server.' and a note: 'A valid email server is required for sending ShakeCast notifications. Leave the username and password fields blank if no authentication is required.' The form contains the following fields:

SMTP Server	Security	Port	Username	Password
<input type="text" value="SMTP Server"/>	<input type="text" value="Security"/>	<input type="text" value="Port"/>	<input type="text" value="Username"/>	<input type="text" value="Password"/>

Notif.

Default Email Template	Default Script Template	From	Envelope From
<input type="text" value="default.txt"/>	<input type="text" value="default.pl"/>	<input type="text" value="shakecast@usgs.gov"/>	<input type="text" value="shakecast@usgs.gov"/>

Figure 2. SMTP Email server configuration via the web interface.

The administrator must enter a valid email address into the **From** and **EnvelopeFrom** fields as the sender email address of the ShakeCast system. The corresponding section inside the sc.conf file is listed as the Notification block as shown below:

```
<Notification>
  SmtplibServer    smtp.gmail.com
  Security         SSL
  Port             465
  DefaultEmailTemplate  default.txt
```

```
DefaultScriptTemplate    default.pl
Password
Username
EnvelopeFrom
From
</Notification>
```

After saving email server information, the administrator will need to restart the ShakeCast notification service in order to reload the new configuration settings.

3.3 Restarting System Services

The ShakeCast administrator requires local administration privileges to restart services from the command line. Restarting system services is needed any time changes are made to entries under the **General Settings** tab or the system configuration file “**sc.conf**.”

Linux: Two shell scripts start/stop daemon services,

- To start services, execute the script “/usr/local/shakecast/sc/rc.d/sc-start-watcher.sh”
- To stop services, execute the script “/usr/local/shakecast/sc/rc.d/sc-stop.sh”

Windows: Two batch files start/stop system services

- To start services, execute the batch “<sc_home>/admin/start_sc_services.bat”
- To stop services, execute the batch “<sc_home>/admin/stop_sc_services.bat”

3.4 Database Management

The ShakeCast database repository tracks information from several different sources, including:

- User provided inventory for facilities, user groups and users.
- Earthquake information and related products including ShakeMap and others from the USGS.
- Local ShakeCast products generated as part of the earthquake processing.
- Notification delivery records and miscellaneous system messages.

Once processed, the above information becomes structured data and may be stored among several interconnected tables inside the database. The figure below illustrates the database schema for user’s facility inventory on basic information, attributes, features, and fragility. The ShakeCast administrator should not edit records of the ShakeCast database directly with a database viewing/editing tool but via the ShakeCast administrative interface or command line to avoid database corruption.

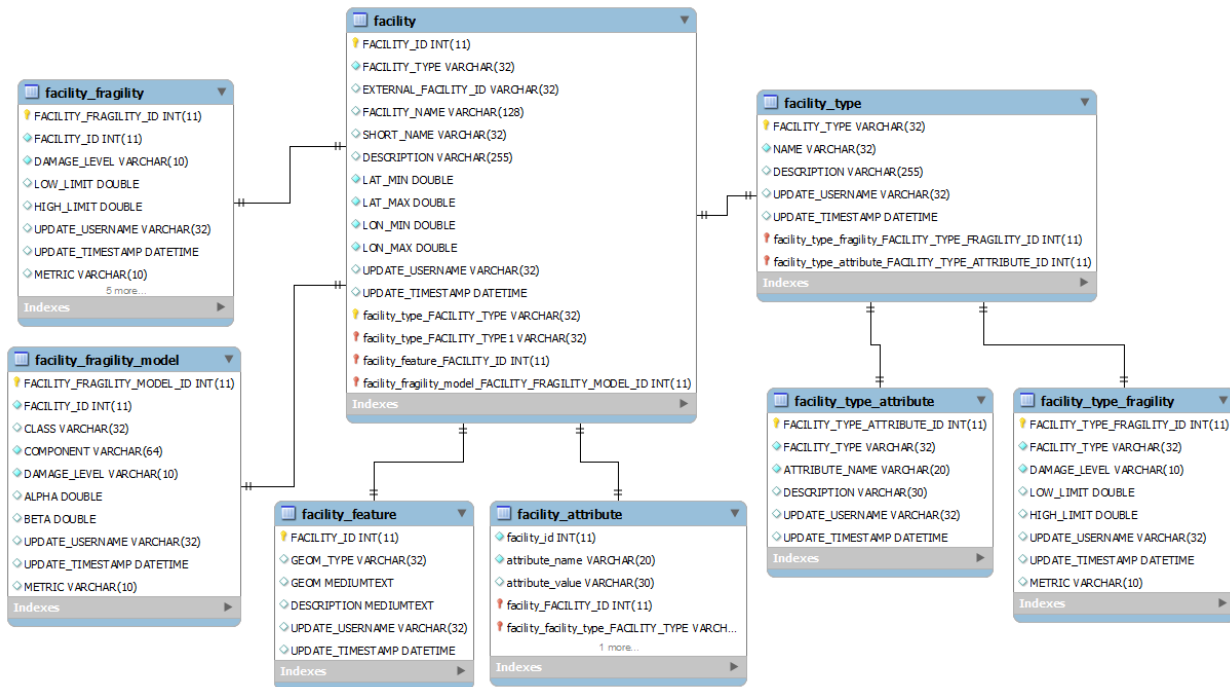


Figure 3. ShakeCast facility inventory database schema

3.5 System and Database Reset

The ShakeCast system’s database and file system will increase as it processes more earthquakes, which may slow the system’s performance, depending on the available resources. Although ShakeCast V3 automatically deletes unwanted events from the system, an administrator can delete all events by resetting the system to its original state.

3.5.1 System Reset

Linux

The administrator must manually purge the database and then delete the installed code base from the file system. After ShakeCast is removed from the system, the administrator can check out a new copy of ShakeCast from the USGS github repository below to perform manual installation.

<https://github.com/klin-usgs/ShakeCast>

Windows

The administrator can use the ShakeCast Installer program to re-install the application to reset a Windows ShakeCast system.

3.5.2 Database Reset

A database reset can manage out-of-control inventory, a critical misconfiguration (e.g., server information) or a ShakeCast database crash.

The database reset is performed from the command line with local Administrator privileges. Switch to the ShakeCast administration script directory default “C:\ShakeCast\admin” and execute the following batch scripts in sequence:

1. Drop existing “sc” database and create a new one: **create_sc_db.bat**
2. Create ShakeCast database tables: **create_sc_tables.bat**
3. Load default data: **load_sc_data.bat**
4. Restart ShakeCast services: **start_sc_services.bat**
5. Activate routine tasks: **inject_init.bat**

```
C:\Shakecast\admin>create_sc_db.bat
Creating ShakeCast database schema...
Enter password: *****
Done.
C:\Shakecast\admin>create_sc_tables.bat
Creating ShakeCast Tables...
Warning: Using a password on the command line interface can be insecure.
Done.
C:\Shakecast\admin>load_sc_data.bat
Loading ShakeCast Data...
Warning: Using a password on the command line interface can be insecure.
Done.
C:\Shakecast\admin>start_sc_services.bat
The requested service has already been started.

More help is available by typing NET HELPMMSG 2182.

The Apache2.2 service is starting..
The Apache2.2 service was started successfully.

The ShakeCast Dispatcher service is starting.
The ShakeCast Dispatcher service was started successfully.

The ShakeCast Notification Generator service is starting.
The ShakeCast Notification Generator service was started successfully.

The ShakeCast Notification Distributor service is starting.
The ShakeCast Notification Distributor service was started successfully.
C:\Shakecast\admin>inject_init.bat
2014-07-25 21:40:35
C:\Shakecast\admin>
```

Figure 4. Database reset sequence in ShakeCast for Windows

3.6 Facility Inventory

To support component-based fragility assignment and geometric footprints for a facility, an XML file format has been defined to accommodate the expanded scope of information.

The figure below shows an example with information for a typical bridge with 20 components. The combined **FACILITY_TYPE** and **FACILITY_NAME** fields must be unique for each facility. Basic information for the facility information is shown in (a). (b) shows geometric features and optional custom html snippet for web presentation. The minimum bounding box of the facility will be used to assess ground motion estimates at the location of facility. (c) shows fragility settings for each defined component and the fragility settings for **SYSTEM** components and will become the representative setting for the purpose of notifications. Fragility settings for each component consist of one chosen metric and one mean and beta pair for every inspection priority.

The fragility settings for HAZUS (FEMA, 2006) model building types are included in ShakeCast, so buildings with HAZUS model building types defined will have fragility settings associated with them in the ShakeCast system.

If an organization does not have comprehensive facility information, the administrator can still use the legacy file format (in CSV) for facility inventory. Details for preparation of facility inventory in the CSV format is described in Appendix D.

(a)

	A	B	C	D	E	F	G
	EXTERNAL_FACILITY_ID	FACILITY_TYPE	COMPONENT_CLASS	COMPONENT	FACILITY_NAME	SHORT_NAME	DESCRIPTION
1	123456	BRIDGE	1_System	SYSTEM	DEMO BRIDGE	123456	Demonstration bridge
2	123456	BRIDGE	2_Primary_TESTING	ABUT_SEAT			
3	123456	BRIDGE	2_Primary_TESTING	COLUMN			
4	123456	BRIDGE	2_Primary_TESTING	HINGE_SEAT			
5	123456	BRIDGE	2_Primary_TESTING	SUPPORT_SEAT			
6	123456	BRIDGE	3_Secondary_TESTING	ABUT_BEARING			
7	123456	BRIDGE	3_Secondary_TESTING	ABUT_RESTRAINER			
8	123456	BRIDGE	3_Secondary_TESTING	ABUT_SEAL			
9	123456	BRIDGE	3_Secondary_TESTING	HINGE_BEARING			
10	123456	BRIDGE	3_Secondary_TESTING	HINGE_RESTRAINER			
11	123456	BRIDGE	3_Secondary_TESTING	HINGE_SEAL			
12	123456	BRIDGE	3_Secondary_TESTING	SUPPORT_BEARING			
13	123456	BRIDGE	3_Secondary_TESTING	SUPPORT_RESTRAINER			
14	123456	BRIDGE	3_Secondary_TESTING	SUPPORT_SEAL			
15	123456	BRIDGE	4_General_Distress_TESTING	ABUTMENT			
16	123456	BRIDGE	4_General_Distress_TESTING	FOUNDATION			
17	123456	BRIDGE	4_General_Distress_TESTING	SHEAR_KEY			
18	123456	BRIDGE	4_General_Distress_TESTING	RESPONSE			
19	123456	BRIDGE	5_Ground_Failure_TESTING	LANDSLIDE			
20	123456	BRIDGE					

(b)

	A	B	I	J
	EXTERNAL_FACILITY_ID	FACILITY_TYPE	FEATURE:GEOM	FEATURE:DESCRIPTION
1	123456	BRIDGE	-124.055065,41.553771,0	<table border="0" cellpadding="3" cellspacing="3" height="250" width=
2	123456	BRIDGE		
3	123456	BRIDGE		
4	123456	BRIDGE		
5	123456	BRIDGE		
6	123456	BRIDGE		
7	123456	BRIDGE		
8	123456	BRIDGE		
9	123456	BRIDGE		
10	123456	BRIDGE		
11	123456	BRIDGE		
12	123456	BRIDGE		
13	123456	BRIDGE		
14	123456	BRIDGE		
15	123456	BRIDGE		
16	123456	BRIDGE		
17	123456	BRIDGE		
18	123456	BRIDGE		
19	123456	BRIDGE		
20	123456	BRIDGE		

(c)

	A	B	K	L	M	N	O	P	Q	R	S	T	U
	EXTERNAL_FACILITY_ID	FACILITY_TYPE	METRIC	METRIC:ALPHA:G	METRIC:BETA:GR	METRIC:ALPHA:G	METRIC:BETA:GR	METRIC:ALPHA:Y	METRIC:BETA:YEL	METRIC:ALPHA:O	METRIC:BETA:OR	METRIC:ALPHA:R	METRIC:BETA:RE
1	123456	BRIDGE	PSA10	REY	EY	REEN	EEN	ELLOW	LOW	RANGE	ANGE	ED	D
2	123456	BRIDGE	PSA10	0.001	0.6	10	0.6	96.94	0.6	116.33	0.6	164.79	0.6
3	123456	BRIDGE	PSA10	0.001	0.6	9.15	0.6	83.37	0.6	94.22	0.6	145.02	0.6
4	123456	BRIDGE	PSA10	0.001	0.6	9.86	0.6	89.18	0.6	100.04	0.6	158.20	0.6
5	123456	BRIDGE	PSA10	0.001	0.6	8.6	0.6	86.27	0.6	104.69	0.6	138.43	0.6
6	123456	BRIDGE	PSA10	0.001	0.6	8.41	0.6	95.00	0.6	101.20	0.6	149.96	0.6
7	123456	BRIDGE	PSA10	0.001	0.6	8.92	0.6	91.12	0.6				
8	123456	BRIDGE	PSA10	0.001	0.6	9.07	0.6	86.27	0.6				
9	123456	BRIDGE	PSA10	0.001	0.6	9.58	0.6	90.15	0.6				
10	123456	BRIDGE	PSA10	0.001	0.6	9.25	0.6	89.18	0.6				
11	123456	BRIDGE	PSA10	0.001	0.6	8.02	0.6	95.00	0.6				
12	123456	BRIDGE	PSA10	0.001	0.6	8.02	0.6	77.55	0.6				
13	123456	BRIDGE	PSA10	0.001	0.6	8.74	0.6	84.34	0.6				
14	123456	BRIDGE	PSA10	0.001	0.6	9.94	0.6	90.15	0.6				
15	123456	BRIDGE	PSA10	0.001	0.6	9.19	0.6	95.97	0.6				
16	123456	BRIDGE	PSA10	0.001	0.6	9.29	0.6	81.43	0.6				
17	123456	BRIDGE	PSA10	0.001	0.6	8.7	0.6	96.94	0.6				
18	123456	BRIDGE	PSA10	0.001	0.6	8.7	0.6	87.24	0.6				
19	123456	BRIDGE	PSA10	0.001	0.6	9.34	0.6	96.94	0.6				
20	123456	BRIDGE	PGA	0.001	0.6								

Note: The fragility parameters shown here were randomly generated for testing and development purposes only.

Figure 5. Sample facility information for one bridge FACILITY: (a) Basic bridge information, (b) bridge geometric features, and (c) fragility settings for bridge components.

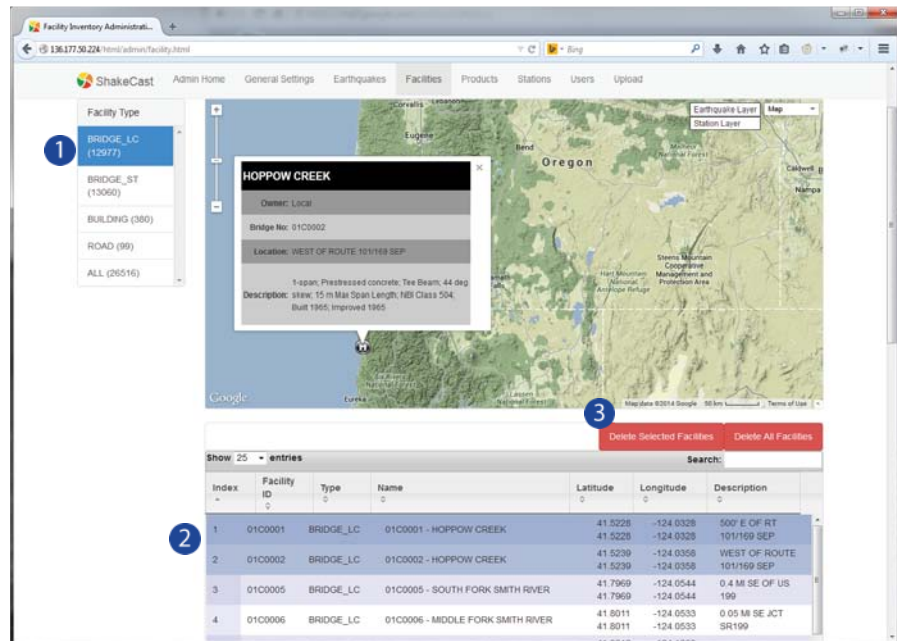
After the facility inventory is saved into a file, e.g., bridge.xml, it can be processed by the ShakeCast system via either the **Upload** tab from the administrative interface or the command line script `manage_facility_xml.pl` with the following syntax

```
C:\ShakeCast\sc\bin\manage_facility_xml.pl bridge.xml
```

To delete selected facilities from the ShakeCast database, select the Facilities tab from the administrative web interface.

1. Select Facility Type
2. Highlight facilities to be deleted
3. Click the **Delete Selected Facilities** button to delete

To delete all facilities of the selected facility type, click the **Delete All Facilities** button



To revert or delete the facility inventory using the tool under the **Upload** tab, drop-in the same facility inventory file and choose the **Delete** option before submit. With the command line script, `manage_facility_xml.pl`, issue the following command:

```
C:\ShakeCast\sc\bin\manage_facility_xml.pl -delete bridge.xml
```

3.7 User Groups

A ShakeCast user GROUP registers both the earthquake monitoring region and the notification requests for the user group. For example, we define an area that covers the entire State of California plus one degree buffer for statewide earthquake monitoring. As the result, the contributing networks for earthquake information are not limited to the California Integrated Seismic Network (CISN) and will also include Pacific Northwest network (UW), the Nevada network (NN), and the Global Seismic Network (GSN). Sub-division groups can be derived from the statewide group to provide notifications for specific needs.

Users can belong to multiple groups to further customize their notification preferences. As an example, the Caltrans State Bridge group configuration file shown below contains four directives:

1. **FACILITY_TYPE** specifies a filter for state bridges only.
2. **POLY** specifies an area for statewide coverage.

3. **NEW_EVENT NOTIFICATION** specifies notification requests for new earthquakes (once per event).
4. **Inspection Priority (DAMAGE) NOTIFICATION** specifies notifications requests for an aggregated list of state bridges in either GREEN, YELLOW, ORANGE, and RED states. This directive will exclude state bridges tagged as GREY (Below Threshold) and will attach a ShakeCast summary PDF file if it is available at the time of notification.

```

<BRIDGE_ST>
  FACILITY_TYPE      BRIDGE_ST

  POLY  43.000 -126.000  \
                39.000 -126.000  \
                34.000 -123.000  \
                31.000 -118.000  \
                31.000 -113.000  \
                36.000 -113.000  \
                39.000 -118.000  \
                43.000 -118.000  \
                43.000 -126.000

  <NOTIFICATION>
    NOTIFICATION_TYPE      NEW_EVENT
    DELIVERY_METHOD        EMAIL_HTML
    EVENT_TYPE              ACTUAL
    AGGREGATE               1
    LIMIT_VALUE             4
  </NOTIFICATION>

  <NOTIFICATION>
    NOTIFICATION_TYPE      DAMAGE
    DELIVERY_METHOD        EMAIL_HTML
    DAMAGE_LEVEL            GREEN
    EVENT_TYPE              ACTUAL
    AGGREGATE               1
    AGGREGATION_GROUP      BRIDGE_ST
    PRODUCT_TYPE            PDF_BRIST
  </NOTIFICATION>

  <NOTIFICATION>
    NOTIFICATION_TYPE      DAMAGE
    DELIVERY_METHOD        EMAIL_HTML
    DAMAGE_LEVEL            YELLOW
    EVENT_TYPE              ACTUAL
    AGGREGATE               1
    AGGREGATION_GROUP      BRIDGE_ST
    PRODUCT_TYPE            PDF_BRIST
  </NOTIFICATION>

  <NOTIFICATION>
    NOTIFICATION_TYPE      DAMAGE
    DELIVERY_METHOD        EMAIL_HTML
    DAMAGE_LEVEL            ORANGE
    EVENT_TYPE              ACTUAL
    AGGREGATE               1
    AGGREGATION_GROUP      BRIDGE_ST
    PRODUCT_TYPE            PDF_BRIST
  </NOTIFICATION>

```

```

<NOTIFICATION>
    NOTIFICATION_TYPE      DAMAGE
    DELIVERY_METHOD        EMAIL_HTML
    DAMAGE_LEVEL            RED
    EVENT_TYPE              ACTUAL
    AGGREGATE               1
    AGGREGATION_GROUP       BRIDGE_ST
    PRODUCT_TYPE            PDF_BRIST
</NOTIFICATION>
</BRIDGE_ST>

```

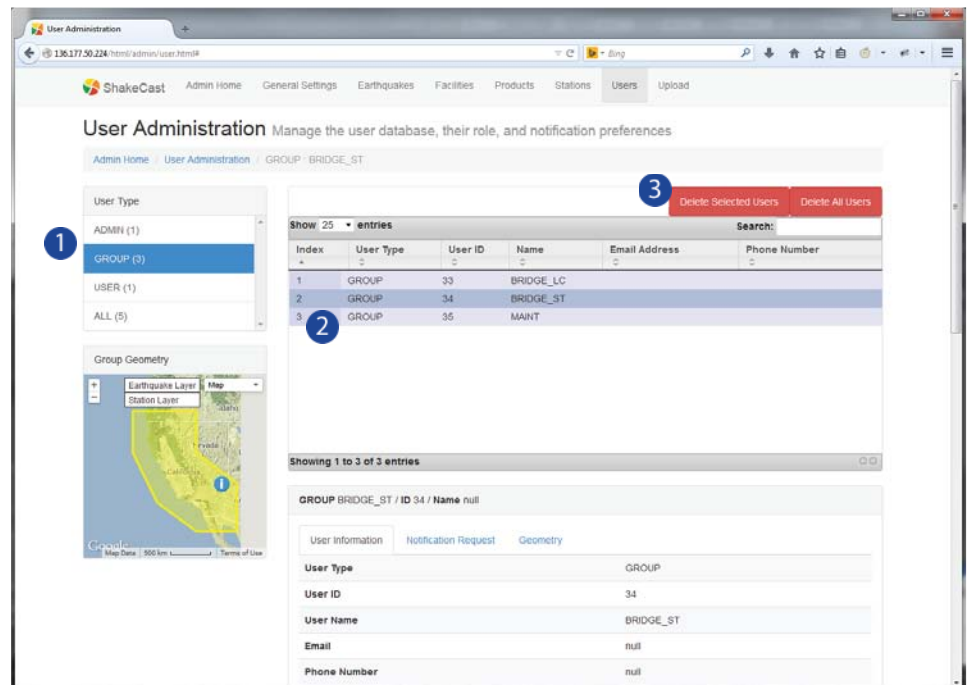
After the group configuration is saved into a file, bridge_st.conf, it can be processed by the ShakeCast system via either the **Upload** tab from the administrative interface or the command line script manage_group.pl with the following syntax

```
C:\ShakeCast\sc\bin\manage_group.pl -conf bridge_st.conf
```

To delete selected groups from the ShakeCast database, select the **Users** tab from the administrative web interface.

1. Select **GROUP** User Type
2. Highlight groups to be deleted
3. Click the Delete Selected Users button to delete

To delete all groups of the selected facility type, click the **Delete All Users** button



To revert or delete the facility inventory using the tool under the **Upload** tab, drop-in the same group inventory file and choose the **Delete** option before submit. With the command line script, manage_group.pl, issue the following syntax

```
C:\ShakeCast\sc\bin\manage_group.pl -delete -conf bridge_st.conf
```


3.8 User Inventory

ShakeCast V3 does not provide user-specific notification preferences and requires a user to sign up for at least one user group in order to receive notifications. When signing up for multiple groups, the group-specific messages will not be aggregated for the user, thus the user may receive several messages with group-specific contents for an earthquake.

In addition to designating user's group and notification preferences, the user inventory also specifies access privileges via the ShakeCast web interface. The defined username and password is for the web access only and does not create local user accounts on the ShakeCast server.

USER_TYPE	USERNAME	PASSWORD	FULL_NAME	EMAIL_ADDRESS	PHONE_NUMBER	DELIVERY:EMAIL_HTML	GROUP:BRIDGE_ST:BRIDGE_LC
USER	klin	sc4all	John Doe	shake@usgs.gov	(123) 456-7890	shake@usgs.gov	BRIDGE_ST

As an example, the above spreadsheet defines the following required information:

1. **USER_TYPE** is either the **USER** or **ADMIN** type. **ADMIN** users have additional privileges for access and ShakeCast administrative web interface.
2. **USERNAME** and **PASSWORD** fields define user access credentials. The combined **USER_TYPE** and **USERNAME** field needs to be unique.
3. **DELIVERY::EMAIL_HTML** field defines the email address for receiving rich content HTML ShakeCast notifications.
4. The **GROUP::BRIDGE_ST::BRIDGE_LC** header field lists the two allowed group designations (**BRIDGE_ST** and **BRIDGE_LC**). The user in the data row has an assigned group for **BRIDGE_ST**.

After the user inventory is saved into a file in the CSV format, *e.g. caltrans_user.csv*, it can be processed by the ShakeCast system via either the **Upload** tab from the administrative interface or the command line script *manage_user.pl* with the following syntax

```
C:\ShakeCast\sc\bin\manage_user.pl caltrans_user.csv
```

Deleting selected users via the administrative web interface is identical to deleting a group. The administrator selects either the **ADMIN** or **USER** type before highlighting users for deletion. To revert or delete the user inventory using the tool under the **Upload** tab, drop-in the same user inventory file and choose the **Delete** option before submit. With the command line script, *manage_user.pl*, issue the following syntax

```
C:\ShakeCast\sc\bin\manage_user.pl -delete caltrans_user.csv
```

3.9 Earthquake and Scenario Inventory

ShakeCast maintains a local earthquake database, effectively a subset of the USGS ShakeMap archive. ShakeMaps for actual earthquakes are received and processed by the system automatically, but scenario

ShakeMaps must be manually triggered by an administrator. Scenarios can be triggered with a scenario ShakeMap package or downloaded directly from the *USGS web site*.

For automated earthquake processing, management of the ShakeCast earthquake inventory for actual events require (1) defining the earthquake monitoring regions; and (2) configuring the triggering and archiving filters.

The filters for earthquake triggering and archiving are optional configurations and are not included under the **General Settings** tab of the administrative interface because of infrequent use. To change the default settings, edit the ShakeCast system configuration file, default at “C:\ShakeCast\sc\conf\sc.conf”.

A snippet of the configuration options shown below dictate the behavior for processing and archiving of actual earthquakes.

- **MAG_CUTOFF** option specifies the minimum magnitude requirement for triggering the ShakeCast process.
- **ARCHIVE_MAG** option specifies minimum magnitude requirement for the earthquake to be permanently archive after the active time window expires.
- **TIME_WINDOW** option specifies the time window (in days) after the origin for an earthquake to be considered as active.

```
MAG_CUTOFF      3
ARCHIVE_MAG     5.0

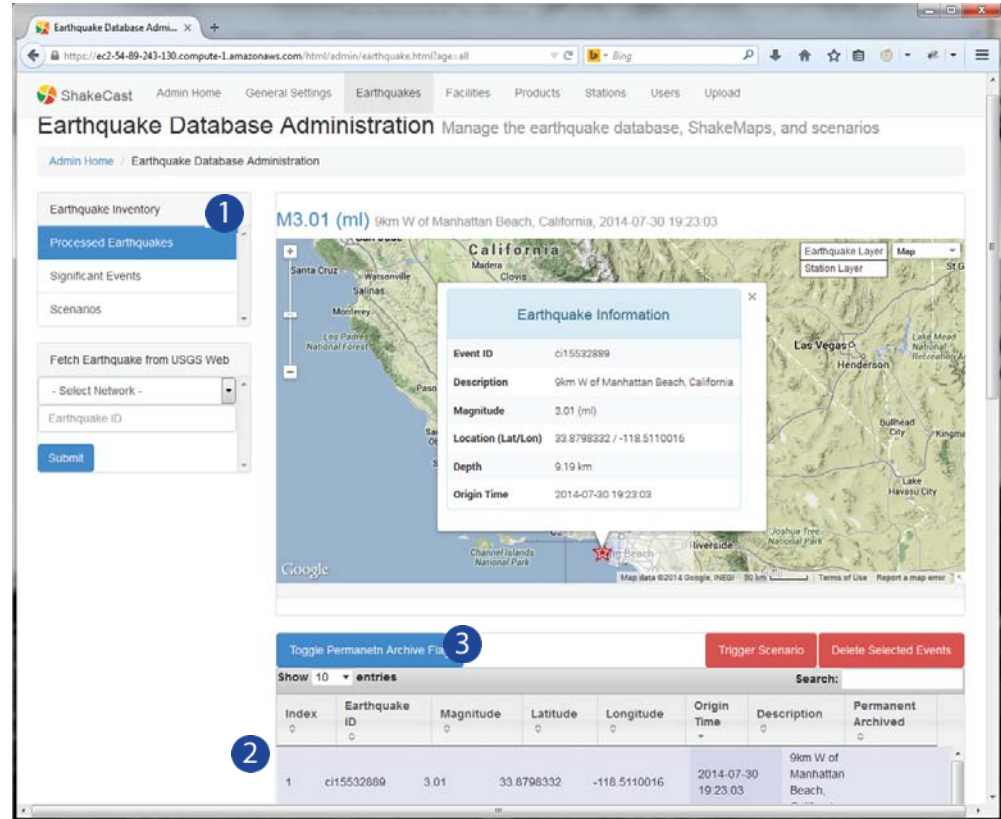
<rss>
  AUTOSTART      1
  TIME_WINDOW    7
  PROMPT         rssid>
  MSGLEVEL       2
  SERVICE_NAME   rssid
  POLL           60
  SPOLL          10
  LOG            C:/Shakecast/sc/logs/sc.log
  REGION         ALL
  SERVICE_TITLE   ShakeCast RSS Daemon
  LOGGING        1
  PORT           53458
</rss>
```

As part of the automated processing, a cron job for earthquake inventory maintenance runs daily to identify and delete unwanted earthquakes from the system. Actual earthquakes with magnitude below the archiving magnitude are removed from the database automatically. Earthquakes above the archive magnitude but without any exposed facilities will also be deleted.

To override the default system behavior, an administrator can set the archiving flag for earthquakes to be excluded from the system archiving policy.

To toggle archive flag for selected earthquakes, select the **Earthquakes** tab from the administrative web interface.

1. Select **Processed Earthquake** list
2. Highlight earthquakes to be archived
3. Click the **Toggle Permanent Archive Flag** button to enable/disable the archive flag



Archived earthquakes are indicated in the **Permanent Archived** column.

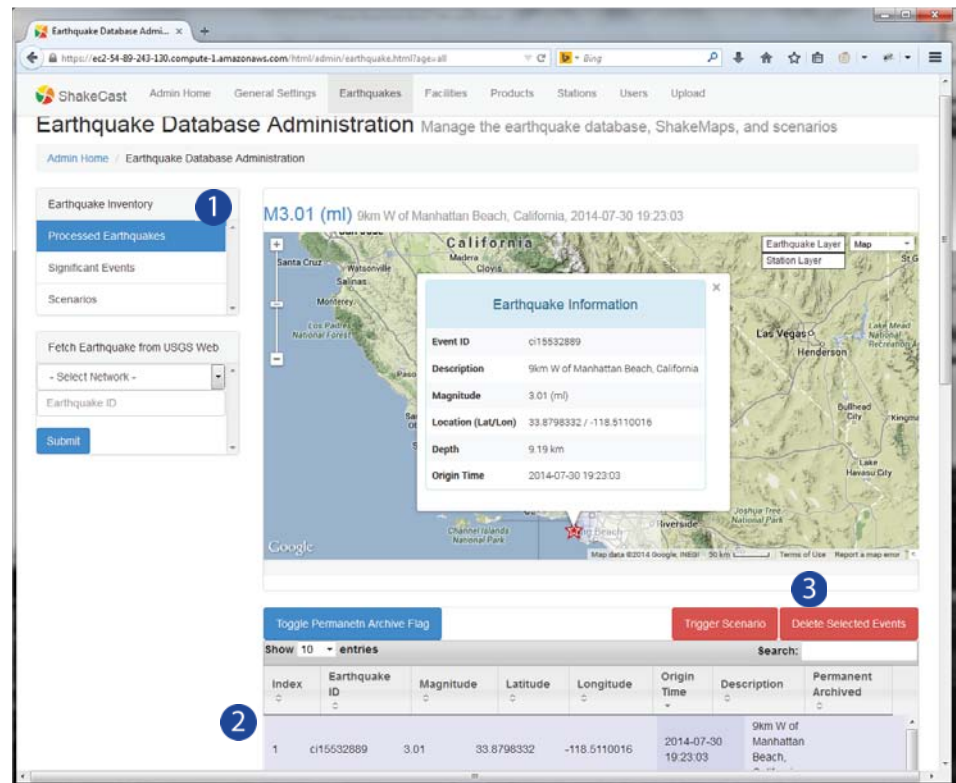
The USGS has prepared ~250 ShakeMap scenario packages for California (80 Northern California and 170 Southern California). To add scenario ShakeMaps to the ShakeCast system, the administrator either upload a premade scenario package via the **Upload** tab from the administrative interface or download the scenario directly from the USGS web site. The command line script `shake_fetch.pl` performs the same function using the following syntax

```
C:\ShakeCast\sc\bin\shake_fetch.pl -network <netid> -event <eventid> -force_run -scenario
```

Both actual and scenario ShakeMaps can be deleted from the ShakeCast system using the administrative interface. To delete selected earthquakes from the ShakeCast database:

Select the **Earthquakes** tab from the administrative web interface.

1. Select Earthquake Inventory
2. Highlight earthquakes to be deleted
3. Click the **Delete Selected Events** button to delete



To revert or delete the facility inventory using the tool under the **Upload** tab, drop-in the same group inventory file and choose the **Delete** option before submit. With the command line script, `manage_group.pl`, issue the following syntax

```
C:\ShakeCast\sc\bin\manage_event.pl -delete <eventid ...>
```

3.10 Notification and System Records

ShakeCast tracks system activities related to executed tasks, background processes, notifications, and state-of-health warning and error messages. General activities are logged into separate files

- `sc.log` stores general information and results of program execution
- `sc_access.log` stores information on access to the system over the web
- `sc_error.log` stores information when programs return with non-normal status

More critical information related to state of notifications and error messages for the system are stored inside the database in two separate tables as a document trail.

To minimize the need for record maintenance, a cron job for log file maintenance runs daily to rotate log files in a ring buffer. The administrator specifies options for the maintenance script inside the system configuration file, “C:\ShakeCast\sc\conf\sc.conf”

```
<Logrotate>
  LOGSTATDIR    C:/Shakecast/sc/images
  rotate-time   1 week
  compress      Yes
  keep-files    5
  status-file    C:/Shakecast/sc/logs/logrotate.status
  max_size      100 M
  logfile       C:/Shakecast/sc/logs/sc.log
  logfile        C:/Shakecast/sc/logs/sc_access.log
  logfile        C:/Shakecast/sc/logs/sc_error.log
</Logrotate>
```

Key fields of the configuration section **Logrotate** from the system configuration are described below:

- **rotate-time** specifies the length in time before the log entries to be removed from the log file
- **keep-files** specifies the number of archived log files to be saved before permanently deleted from the system
- **max_size** specifies the maximum allowed file size of the log files. Once the file size exceeds the limit, log entries will be removed even before the rotation time is reached
- **logfile** specifies the list of log files to be maintained and rotated

Maintenance of system records stored inside the ShakeCast database is usually not required and is only needed when the system shows signs of slow performance, such as delayed notifications. Accumulation of records depends primarily on the number of inventory and notifications configured for a particular system and can vary greatly from one system to another. The administrator should not interact with the database directly and instead execute the command to clean up stored messages in the database at a suggested frequency of once per quarter:

```
C:\ShakeCast\sc\util\clear_notify_table.pl
```

4 ShakeCast End User Customization

ShakeCast delivers post-earthquake and inspection prioritization information to users in several different formats, including:

- Email notifications
- ShakeCast Summary Report
- ShakeCast XML, GoogleEarth KML files and Excel spreadsheets (ShakeCast Local Products)

In addition, ShakeCast users can interact with the system to retrieve detailed results of the analysis, including:

- ShakeCast Website
- ShakeCast Application Programming Interface (API)

In this section we describe the default templates, products, and the ShakeCast web interface. We also show the procedure to customize the above contents to improve user experience. Refer the ShakeCast Technical Guide for detailed description on the ShakeCast local product and API.

4.1 ShakeCast Email Notifications

Email is the primary delivery method of ShakeCast notifications following an earthquake. For most users, email notifications provide a sufficient amount of detail about the event, the impact on facilities and may be the only ShakeCast product used.

ShakeCast is pre-configured to send four types of emails: NEW EVENT, UPDATED EVENT, CANCELED EVENT, and FACILITY ASSESSMENT. The type of message is in the subject line.

ShakeCast notifications are template-driven to meet users' specific requirements and the Administrator can modify the default template and to create new ones for specific user groups. Notifications are generated in real time and their appearance may vary with different email clients, so should be thoroughly tested.

To address access problems from outside the organization's network, earthquake products and images stored on the internal server can be included as attachments to email notifications. The administrator does this by editing the notification templates. Any file available on the local ShakeCast server can be attached to a notification, but the administrator needs to consider the overall size of email notifications on performance.

ShakeCast typically generates email notifications within one or two minutes from receipt of ShakeMap data from the USGS and ShakeMap data is typically generated within 5 minutes of the event. There have been events over the course of the project where longer ShakeMap generation times were observed, which results in longer times for users to receive ShakeCast email notifications.

4.2 Default Facility Assessment Message

The ShakeCast system is triggered when an earthquake with a magnitude of 3.0 or greater is reported by the USGS. Inspection Priority emails are sent if the estimated peak ground motion (PGM) is greater than the triggering threshold. This email includes the list of facilities of specified type and their Inspection Priority.

Facility assessment messages contain the following key features:

- A summary of the event with date, time, magnitude, epicenter and location.
- Key shaking metrics for the ShakeMap.
- A graphic image of the ShakeMap showing the extent of strong shaking.
- The complete list of facilities analyzed and their relative Inspection Priority.
- An optional attachment of ShakeCast Report in PDF (depending on availability at the time of notification) showing detailed information.

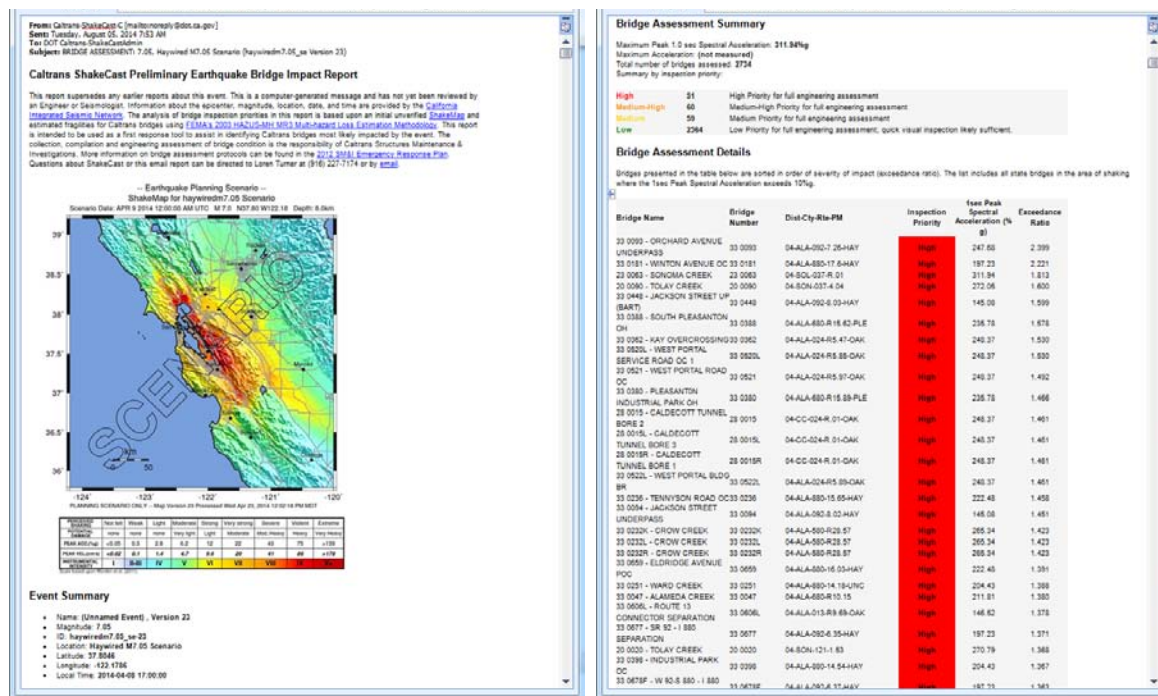


Figure 6. ShakeCast Facility Assessment message and contents.

4.3 Notification Template

All ShakeCast notifications and products are template-driven to meet users' specific requirements. The default template for facility assessment message shown in the figure above is described in details in this section. The administrator can modify the default template and/or create new templates for specific user groups.

4.3.1 Template Directory

ShakeCast is pre-configured to send four types of emails: NEW EVENT, UPDATED EVENT, CANCELED EVENT, and FACILITY ASSESSMENT. The type of message is in the subject line.

ShakeCast notifications are template-driven to meet users' specific requirements and the Administrator can modify the default template and to create new ones for specific user groups. Notifications are generated in real time and their appearance may vary with different email clients, so should be thoroughly tested.

The template directory path is a directive in the system configuration file, `sc.conf`, e.g.,

```
TemplateDir    C:/Shakecast/sc/templates
```

Email notification templates are organized by notification type, then the message type in the template directory.

TemplateDir > Notification Type Directory > Message Type Directory

Valid notification types and their triggers are:

- **New Event** (`new_event`): a new earthquake is declared
- **Updated Event** (`upd_event`): an earthquake's source parameters change
- **Canceled Event** (`can_event`): an earthquake is canceled
- **Facility Assessment** (`damage`): fragility assessed at a facility for an event's ground motion measurements
- **New Product** (`new_prod`): a new registered earthquake product becomes available
- **Facility Shaken** (`shaking`): ground shaking estimates are available at a single facility
- **System** (`system`): the system's encountered an error during execution

4.3.2 Customizing Email Notifications

The notification template consists of either one or multiple files, depending the message aggregation option.

- If the aggregation flag is not set in the notification request, a single template file is used.
- If the aggregate flag is set, the template file is divided into header, body and footer segment files. Among the three files, the body segment aggregates notifications (event, product, shaking, damage, and system) into a table and the table is sorted based on directives in a separate configuration file.

The filename "default" is used when no custom notification template is requested e.g., for a non-aggregated template (damage notification) the content of the template shown below will be saved into a single file, *default.txt*. The aggregated template consists of three template segments, *default_header.txt*, *default_body.txt*, and *default_footer.txt*


```

; Default HTML Email Notification -- header
;
From: ShakeCast <%HEADER_FROM%>
To: %HEADER_TO%
Subject: Inspection Priority M%MAGNITUDE% - %EVENT_LOCATION_DESCRIPTION% (%EVENT_ID%)
Attach: C:/ShakeCast/sc/data/%SHAKEMAP_ID%-%SHAKEMAP_VERSION%/intensity.jpg
; End -- header

<font size=+2><b>ShakeCast Event: Magnitude %MAGNITUDE%</b></font><br>
ShakeMap %EVENT_NAME:|NULL|;(Unnamed Event)% Version %SHAKEMAP_VERSION%<br>
Event Location: %EVENT_LOCATION_DESCRIPTION%<br>
Event Time: %EVENT_TIMESTAMP%<br>
Generated at %GENERATION_TIMESTAMP%<br>
Reported by: Server ID = %SERVER_ID%, DNS = %DNS_ADDRESS%
<p>
<font size=+2><b>Damage Summary</b></font><br>
Number of Facilities Reported: %_ITEMNO%<br>
Max Value: MMI: %_MAX_METRIC_MMI%;
Acceleration: %_MAX_METRIC_PGA:|NULL|;(not measured)%<br>
Number of Reports of Likely Damage: %_NUM_DAMAGE_LEVEL_RED%<br>
Number of Reports of Possible Damage: %_NUM_DAMAGE_LEVEL_YELLOW%
<p>
<font size=+2><b>Facility Damage Estimates from ShakeMap</b></font>

<table border=1 bordercolor=black cellpadding=5 cellspacing=0>
<tr>
<th>Facility
<th>Inspection Priority
<th>Metric
<th>Value
<th>Exceedance Ratio
</tr>
; ; ; ;

; Default HTML Email Notification -- body
;
<tr>
<td>%FACILITY_NAME%
<td
bgcolor=%DAMAGE_LEVEL:RED;RED;YELLOW;YELLOW;ORANGE;ORANGE;GREEN;GREEN%>%DAMAGE_LEVEL:RED;High;ORANGE;Moderate High;YELLOW;Moderate;GREEN;Low%
<td>%METRIC%
<td>%GRID_VALUE%
<td>%EXCEEDANCE_RATIO%
</tr>

; Default HTML Email Notification -- footer
;
</table>
[END]

```

Sorting in the body template segment below is done in the file, *default.conf*.

```

;
; ; ; config for email html
;
sort severity_rank/d grid_value/d exceedance_ratio/d facility_name/t
;
; ; ;

```

4.3.3 Template Layout and Content

ShakeCast users receive email notifications based on the design of layout and content for the template rendered in real time. Design of an HTML-based template is similar to a web page and usually requires iterations to get the expected user experience. The result may also vary depending on user's email client.

Attachments: If an organization has security or firewall issues for users outside of their network, an administrator can include one or more earthquake products as email attachments. Any file available on the local ShakeCast server can be included in the notification's header, but the administrator should balance the size of the email notifications with performance goals. Email notification attachment is a new feature of the V3 system and is done with a header directive e.g.,

```
Attach: C:/ShakeCast/sc/data/%SHAKEMAP_ID%-%SHAKEMAP_VERSION%/intensity.jpg
```

Keywords: Keywords enclosed by the “%” symbol in are replaced with their corresponding values stored in the ShakeCast database.

- The scope of keywords varies for the intended notification type. Appendix I lists available keywords for all notification types.
- Keywords for derived values (specific to individual notification message) are prepended with an underscore symbol “_”, e.g., **_MAX_METRIC_MMI**.
- Template-specific value mapping for keywords are separated by the symbol “;”, e.g.,

```
%DAMAGE_LEVEL:RED;High;ORANGE;Moderate High;YELLOW;Moderate;GREEN;Low%
```

maps RED to High, ORANGE to Moderate High, YELLOW to Moderate, and GREEN to Low in the facility damage assessment table.

4.4 ShakeCast PDF Summary Report

The ShakeCast V3 system includes a PDF engine that generates formatted summary reports as standalone products or the PDF report can be sent directly to users as attachments to email notifications. The default PDF report consists of one ShakeCast summary page. The administrator can customize the PDF report similar to customizing email notifications.

The ShakeCast summary report provides:

- A summary of basic earthquake parameters, including origin time, magnitude, hypocenter, and the name of the region where the earthquake took place.
- A map with an intensity overlay showing the extent of shaking, facilities (enlarged icon for facility with shaking estimates) and recent seismicity in the region.
- A bar at the bottom of the map showing color-coded Inspection Priority and the number of facilities in each level.
- A table showing detailed parameters for each assessed facility. The template-driven table fields include the name, epicentral distance, Inspection Priority, PGM measures, and estimated Vs30.

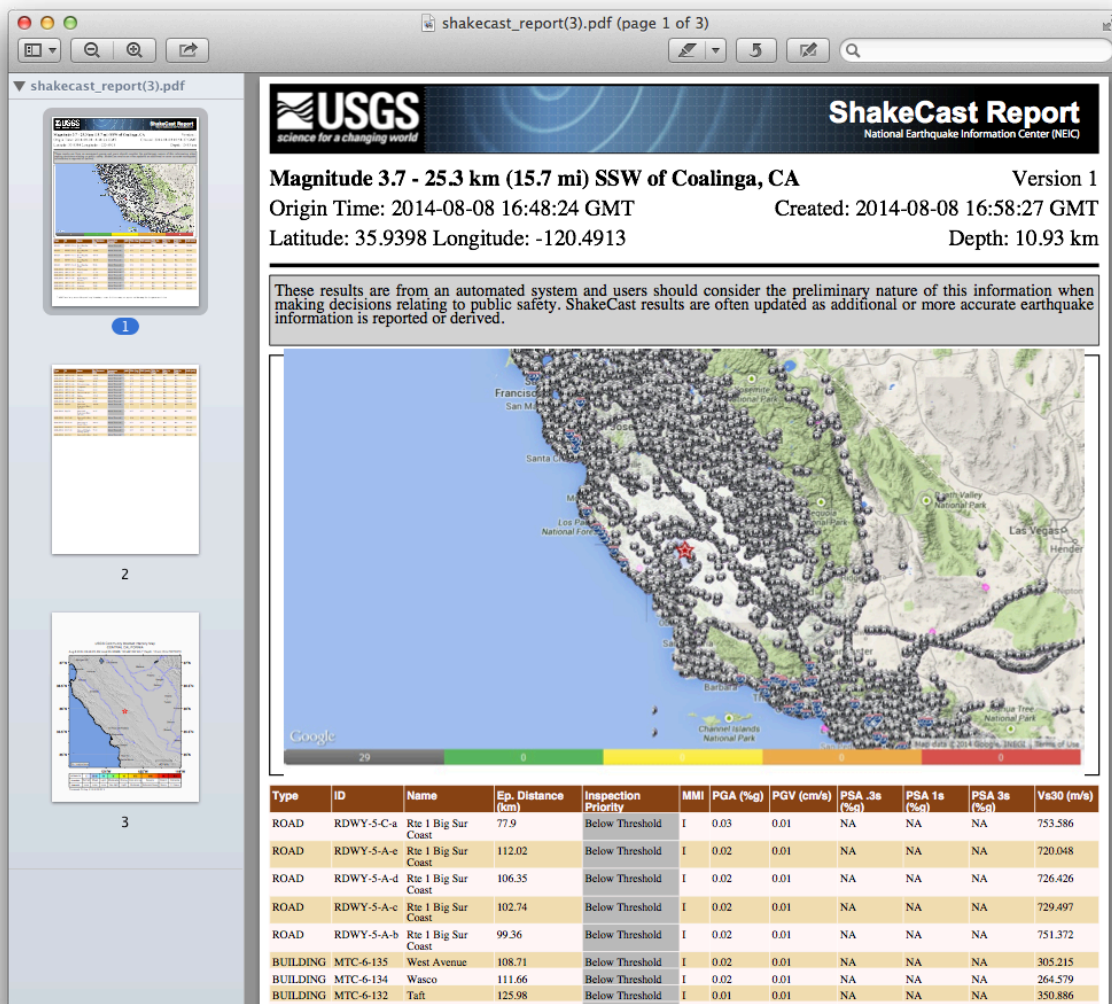


Figure 7. ShakeCast summary page from the PDF summary report.

4.4.1 PDF Summary Report Template

The template-driven ShakeCast PDF report works in the same fashion as email notifications. The administrator can modify the default template and create new ones for specific user groups.

PDF report templates are in the templates directory: e.g.,

C:/Shakecast/sc/templates/pdf/

The ShakeCast PDF report consists of a template file (shakecast_report.pdf) in PDF format and a configuration file (shakecast_report.conf) in XML format. The base PDF template contains static

content in writable PDF format. The administrator can modify the template to reflect their organization's identity.

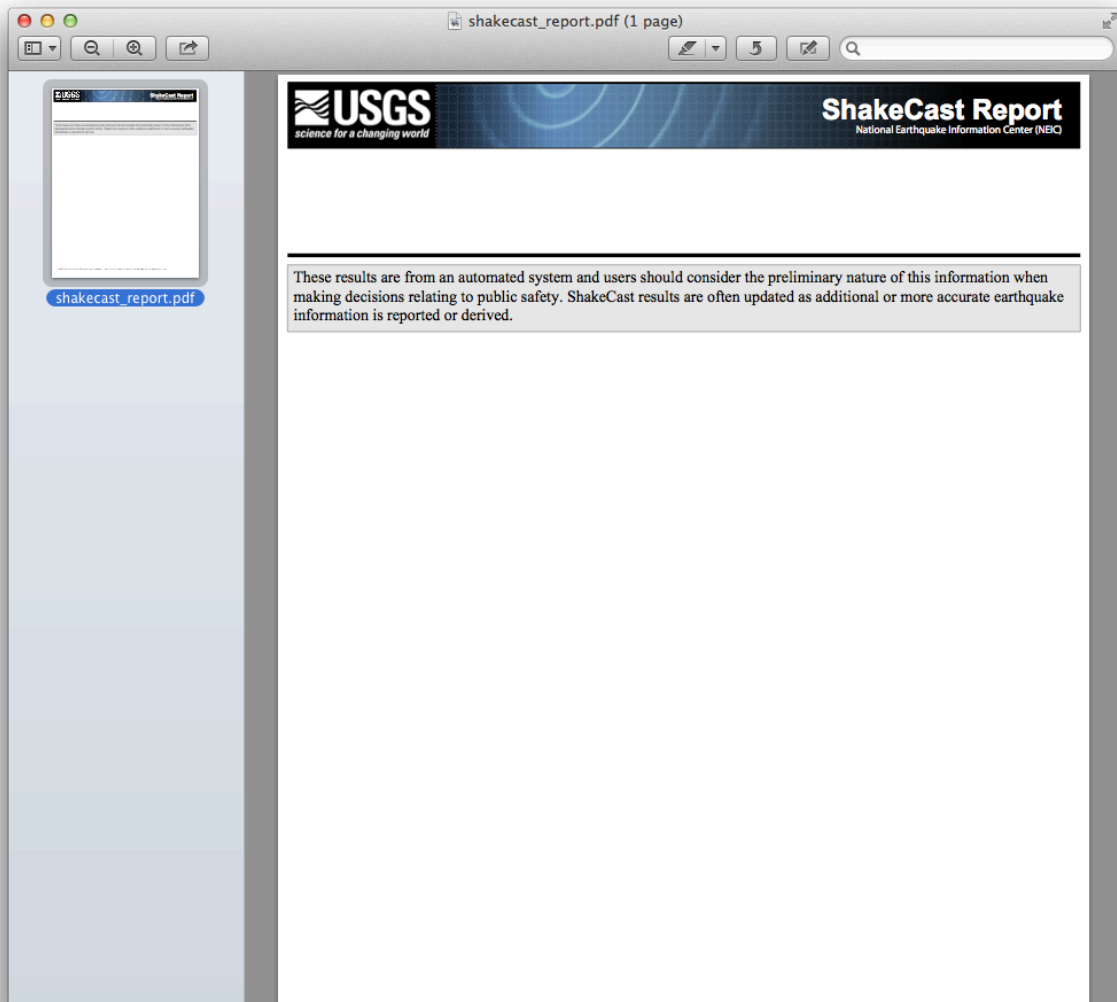


Figure 8. The default template in PDF for the summary report.

4.4.2 Template Layout and Content

The base PDF template above contains static content and is in writable PDF format. The included default template was created using the Adobe Illustrator program. The administrator can modify the template to reflect the identity of user's organization using a PDF editing program.

ShakeCast uses XML tags for common actions of creating PDF files. ShakeCast translates each XML tag in the PDF configuration file and creates one element in the output PDF in the following hierarchy:

<template> --> **<page>** --> **<block>** / **<text>** / **<image>** / **<table>**

- **<template>** declares a PDF template
- **<page>** inserts a new page in the PDF output file with an optional number attribute to specify the page number. The pdf attribute imports an external PDF document into the specified page. The page will be skipped if the requested PDF document is not available.
- **<block>** a general purpose container. It creates and inserts a vector image at the specified size and location. **<block>** tag can enclose all element types to create a complex element.
- **<text>** inserts a string of text into the specified location and bounding box. Line breaks will be inserted and font size will be adjusted automatically to contain the entire string.
- **<image>** inserts an external image in either jpeg, gif, or png format to the specified location and dimension. Note: the image may be distorted if the specified dimension is not proportional to the original image.
- **<table>** reads an existing table in csv format on the local system and creates a formatted table with the specified table fields. The table is inserted to the specified location of the page and the height of the table is determined by the number of entries to be included. New pages will be added to the output PDF file to include the entire table. The table is sorted by the inspection priority (color-coded) then epicentral distance.

Keywords enclosed by the “[]” bracket within XML tags are replaced with their corresponding values stored inside the ShakeCast database. The element location is in the Cartesian coordinate system with the bottom-left corner of the page as the origin.

5 ShakeCast System Administration

The ShakeCast administrative interface is platform independent and is designed for an administrator to perform common tasks ranging from management of facility and user inventory to system-wide maintenance and configuration. Access to the administrative page is restricted to ShakeCast users with administrative privileges. Tasks that can be performed from the interface include:

- General system configuration
- Earthquake/ShakeMap management
- Facility management
- Station management
- Group/User account and notification management
- Inventory upload

The ShakeCast administrative interface does not manage system level services and supporting software: three ShakeCast daemon services (dispatch, notify queue, notify processes), the Apache Web server and MySQL database applications. An administrator must log on to the server system where the ShakeCast system resides to make changes to the configuration files of applications and to start and stop ShakeCast system processes and supporting software.

5.1 Scope of System Management

Most ShakeCast system administrative functions can be accessed via the web interface. The administrative web interface is a collection of HTML5-compatible pages that interact with the backend ShakeCast database via a uniform Application Programming Interface (API). ShakeCast V3 prohibits direct editing (except delete) of inventory from the database. A universal file drop page is created in V3 to receive updated inventory data.

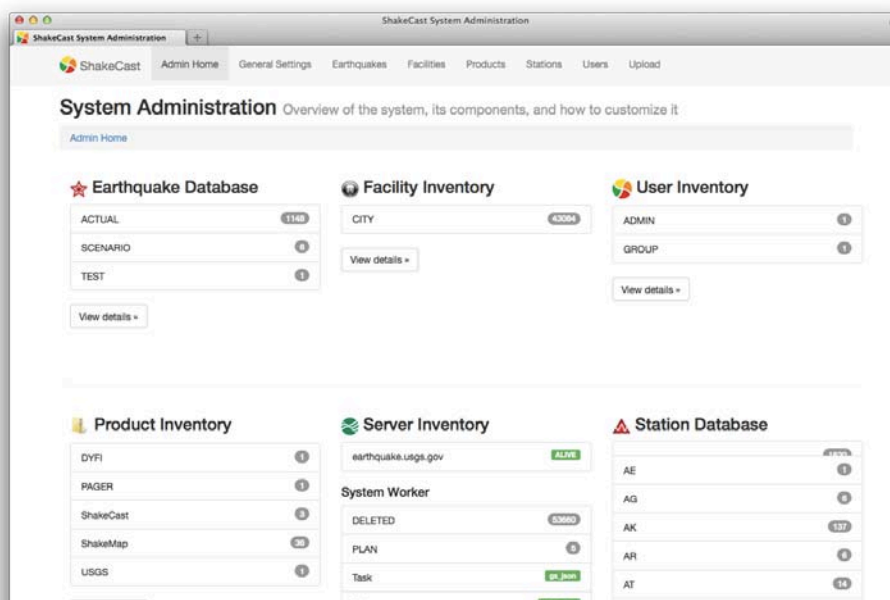


Figure 9. ShakeCast administrative web interface. The default page displays summary of the system inventory stored inside the database and active processes.

5.2 General Settings

The General Settings page allows the ShakeCast administrator to manage system-wide configurable information in six different categories: 1) ShakeCast Database 2) Email Server 3) ShakeMap Server 4) ShakeMap Region 5) System Directory and 6) Misc. Parameter.

NOTE: A system restart is required to load new system configurations.

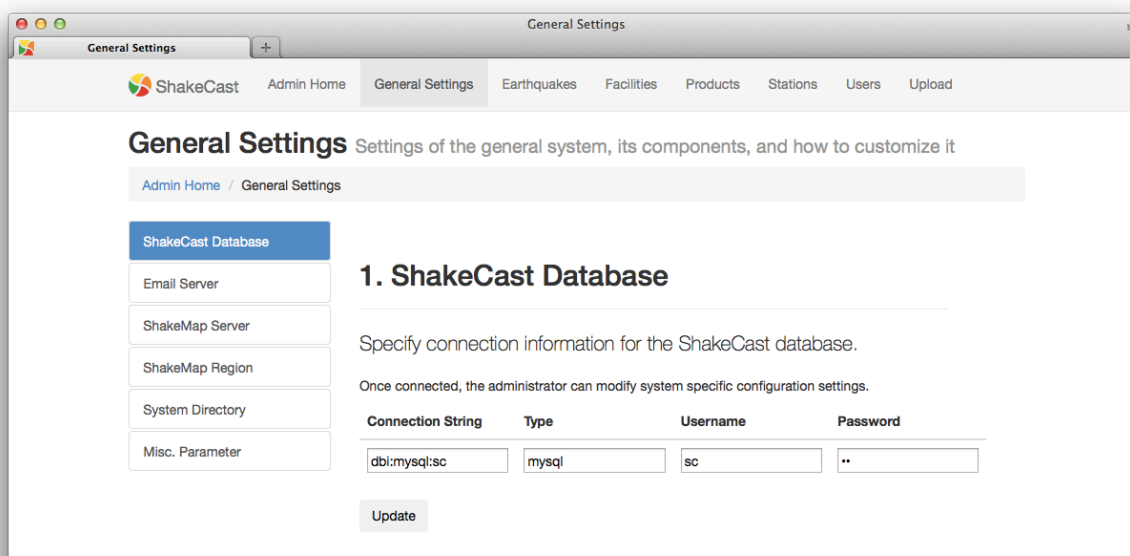


Figure 10. ShakeCast administrative web interface for general system settings.

5.2.1 System Services Restart

The ShakeCast administrator requires local administration privileges to restart services from the command line. As a general rule of thumb, restarting system services is needed any time changes are made to entries under the **General Settings** tab or the system configuration file “**sc.conf**.”

Linux: Two shell scripts start/stop daemon services,

- To start services, execute the script “/usr/local/shakecast/sc/rc.d/sc-start-watcher.sh”
- To stop services, execute the script “/usr/local/shakecast/sc/rc.d/sc-stop.sh”

Windows: Two batch files start/stop system services

- To start services, execute the batch “<sc_home>/admin/start_sc_services.bat”
- To stop services, execute the batch “<sc_home>/admin/stop_sc_services.bat”

5.2.2 ShakeCast Database

The ShakeCast Database form specifies the database connection information. The ShakeCast database account has full privileges to the ShakeCast database only and changes to the database account require database administration privileges. Users should consult with the system administrator and MySQL documentation for information on modifying the database and access privileges.

Table 1. ShakeCast database server information.

Field	Description	Example
Connection String	ShakeCast database name	db:mysql:sc
Type	Database engine	mysql
Username	Username	sc
Password	Password	xx

5.2.3 Configuring the Notification Email Server

ShakeCast V3 supports the following email protocols,

Mail Transport Agent (MTA) via mailx

- SMTP with plain text (port 25) with and without authentication
- SMTP over Secure Sockets Layer (SSL, port 465)
- SMTP over Transport Layer Security (TLS, port 587)

By default, Linux-based ShakeCast installations (e.g., ShakeCast AWS VM) use the built-in mailx utility and require no additional manual email server configuration. However, the administrator can choose to override the default by specifying an SMTP email server. The Windows system typically lacks MTA support, so the administrator is advised to enter an SMTP server for delivering ShakeCast notifications.

The administrator can enter the SMTP information from either the web interface under the **General Settings** tab or by editing the ShakeCast system configuration file located at:

“<sc_home>/sc/conf/sc.conf.”

Required SMTP email server information:

- DNS hostname of the SMTP server
- Security Layer (none/SSL/TLS)
- Service port (25/465/587)
- Username and Password

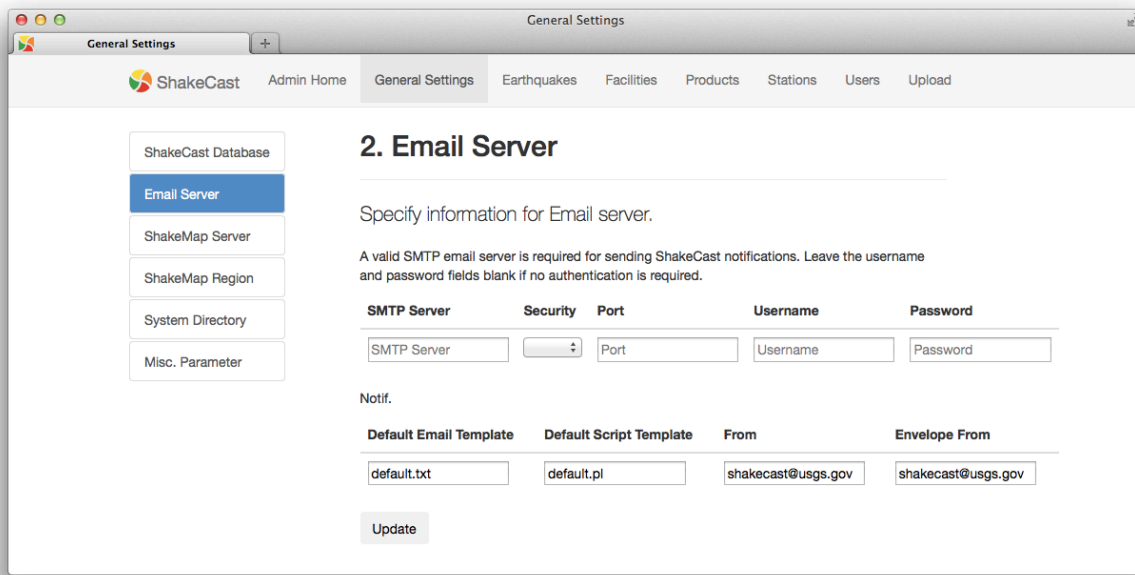


Figure 11. ShakeCast administrative web interface for Email Server configuration.

Table 2. SMTP Email server information.

Field	Description	Example
SMTP Server	SMTP Email server name	smtp.gmail.com
Security	Security option	SSL
Port	Connection port	465
Username	Username	sc
Password	Password	xx
Default Email Template	Default Email template	default.txt
Default Script Template	Default Script template	default.pl
From	Sender email	shakecast@usgs.gov
Envelope From	Reply email	shakecast@usgs.gov

The administrator must enter a valid email address into the **From** and **EnvelopeFrom** fields as the sender email address of the ShakeCast system.

The corresponding section inside the sc.conf file is listed as the Notification block as shown below,

```
<Notification>
    SmtServer      smtp.gmail.com
    Security       SSL
    Port           465
```

```

DefaultEmailTemplate      default.txt
DefaultScriptTemplate     default.pl
Password
Username
EnvelopeFrom             shakecast@usgs.gov
From                      shakecast@usgs.gov
</Notification>

```

After saving email server information, the administrator will need to restart the ShakeCast notification service to reload the new configuration settings.

5.2.4 ShakeMap Server

The ShakeMap Server form specifies ShakeMap server information and is the source of other USGS earthquake products. All ShakeMap products and maps, and the DYFI, PAGER, tectonic summary and historical seismicity plot products are retrieved automatically. The administrator can enable/disable selected servers by toggling the Query flag if multiple servers have been configured.

The screenshot shows the ShakeCast administrative web interface. The top navigation bar includes links for General Settings, Earthquakes, Facilities, Products, Stations, Users, and Upload. The left sidebar contains a menu with options: ShakeCast Database, Email Server, ShakeMap Server (highlighted), ShakeMap Region, System Directory, and Misc. Parameter. The main content area is titled '3. ShakeMap Server' and includes a description: 'Specify server information for ShakeMap and related earthquake products.' Below this, there is a table with three columns: Server ID, DNS Address, and Query. The first row contains the values 1302, earthquake.usgs.gov, and a checked checkbox. An 'Update' button is located below the table. Below the table, there is a section titled '4. ShakeMap Region' with a description: 'Specify region preference for ShakeMap and related earthquake products.'

Figure 12. ShakeCast administrative web interface for ShakeMap Server configuration.

Table 3. Upstream server information.

Field	Description	Example
Server ID	Unique server ID	1302
DNS Address	Server hostname	earthquake.usgs.gov
Query	Active server flag	checked

5.2.5 ShakeMap Region

The ShakeMap Region form specifies information of the ShakeMap regions to be processed by the system. The ShakeMap Region directive functions as a spatial filter based on the predefined boundaries of seismic networks. With the user-defined monitoring region via the GROUP definition, users combine notification requests with user-defined regions to improve performance of the system without unnecessary processing.

For example, users interested in receiving notifications for California enter “NC CI SC NN” corresponding to the Northern California, Southern California, and Nevada networks. The only exception is when there is no group defined in the database and the monitoring regions coincide with the seismic networks.

The screenshot shows the ShakeCast administrative web interface. The top navigation bar includes links for Admin Home, General Settings (selected), Earthquakes, Facilities, Products, Stations, Users, and Upload. On the left sidebar, there are links for ShakeCast Database, Email Server, ShakeMap Server, ShakeMap Region (highlighted), System Directory, and Misc. Parameter. The main content area is titled "4. ShakeMap Region" and contains a description: "Specify region preference for ShakeMap and related earthquake products." Below this is a form with four fields: Region (set to ALL), Active Time Window (set to 7), Magnitude Threshold (set to 3), and Update Threshold (set to 10). An "Update" button is located below the form. Below the form, there is a section titled "5. System Directory" with a description: "Specify region preference for ShakeMap and related earthquake products."

Figure 13. ShakeCast administrative web interface for ShakeMap Region configuration.

Table 4. ShakeMap region configuration information.

Field	Description	Example
Region	ANSS network region code. Multiple values are separated with comma or white space.	NC,SC,UT,NV
Active Time Window	The number of days from origin to trigger ShakeCast process	7
Magnitude Threshold	The minimum magnitude to trigger ShakeCast process.	3.5
Update Threshold	The percent change in PGM to trigger ShakeCast process.	10

5.2.6 System Directory and Miscellaneous Parameters

The System Directory and Miscellaneous Parameters forms specify information of main directories of ShakeCast and its supporting application. Users usually do not need to change the default settings and will need verify access permissions (data directory needs to be readable from the Internet) when specifying a new location in the file system.

The screenshot shows the 'General Settings' page of the ShakeCast web interface. The left sidebar contains a menu with options: ShakeCast Database, Email Server, ShakeMap Server, ShakeMap Region, System Directory (highlighted), and Misc. Parameter. The main content area is titled '5. System Directory' and includes a description: 'Specify region preference for ShakeMap and related earthquake products.' Below this, there are three input fields for 'Root Directory', 'Data Root', and 'Template Directory', all containing the path '/usr/local/shakecast/sc'. There are also input fields for 'Log Directory' (containing '/usr/local/shakecast/sc'), 'Log File' (containing 'sc.log'), and 'Log Level' (set to '2'). An 'Update' button is located at the bottom of the form.

Root Directory	Data Root	Template Directory
/usr/local/shakecast/sc	/usr/local/shakecast/sc	/usr/local/shakecast/sc

Log Directory	Log File	Log Level
/usr/local/shakecast/sc	sc.log	2

The screenshot shows the 'General Settings' page of the ShakeCast web interface, specifically the '6. Misc. Parameters' section. The left sidebar is the same as the previous screenshot, but 'Misc. Parameter' is highlighted. The main content area is titled '6. Misc. Parameters' and includes the same description: 'Specify region preference for ShakeMap and related earthquake products.' Below this, there are three input fields for 'GNU Plot', 'wkhtmltopdf', and 'Perl', all containing the path '/usr/bin/gnuplot', '/usr/local/shakecast/w', and '/usr/local/shakecast/sc' respectively. There are also input fields for 'User ID' (containing 'apache'), 'Group ID' (containing 'apache'), and 'Redundancy Check' (set to '0'). An 'Update' button is located at the bottom of the form.

GNU Plot	wkhtmltopdf	Perl
/usr/bin/gnuplot	/usr/local/shakecast/w	/usr/local/shakecast/sc

User ID	Group ID	Redundancy Check
apache	apache	0

Figure 14. ShakeCast administrative web interface for System Directory (top) and Misc. Parameters configuration (bottom).

Table 5. Directory and Miscellaneous Parameters.

Field	Description	Example
Root Directory	ShakeCast install directory	/usr/local/shakecast/sc
Data Root	ShakeCast data directory	/usr/local/shakecast/sc/data
Template Directory	ShakeCast template directory	/usr/local/shakecast/sc/template
Log Directory	ShakeCast log directory	/usr/local/shakecast/sc/logs
Log File	ShakeCast log file	sc.log
Log Level	ShakeCast log level	2
GNU plot	Path to gnuplot	/usr/bin/gnuplot
wkhtmltopdf	Path to wkhtmltopdf	/usr/bin/wkhtmltoimage
Perl	Path to perl	/usr/bin/perl
User ID	User ID for ShakeCast process (Linux only)	www
Group ID	Group ID for ShakeCast process (Linux only)	www
Redundancy Check	Best effort to detect event under different ID	0

5.3 Earthquake Database Management

The Earthquake Database Management section allows a ShakeCast administrator to manage earthquake inventory in three different categories:

- Processed ShakeMap events
- Significant earthquakes list
- ShakeMap scenarios

Processed ShakeMap events are those automatically downloaded and processed by the user's system based on the ShakeMap Region configuration. Events flagged by the user as "significant" can be more readily accessed. Scenarios are hypothetical event ShakeMaps produced and stored on the USGS web pages:

<http://earthquake.usgs.gov/earthquakes/shakemap/list.php?x=1&s=1>

ShakeCast users can "inject" (import) any scenarios for their own for ShakeCast testing or for earthquake planning exercises using ShakeMap/ShakeCast.

The web form inside the page is used to retrieve additional ShakeMap inventory from the USGS Web site. The ShakeCast local test event type "*_scte" is merged with the scenario type "*_se" in V3. The scenario is supported but users are encouraged to switch to using the standard scenario type.

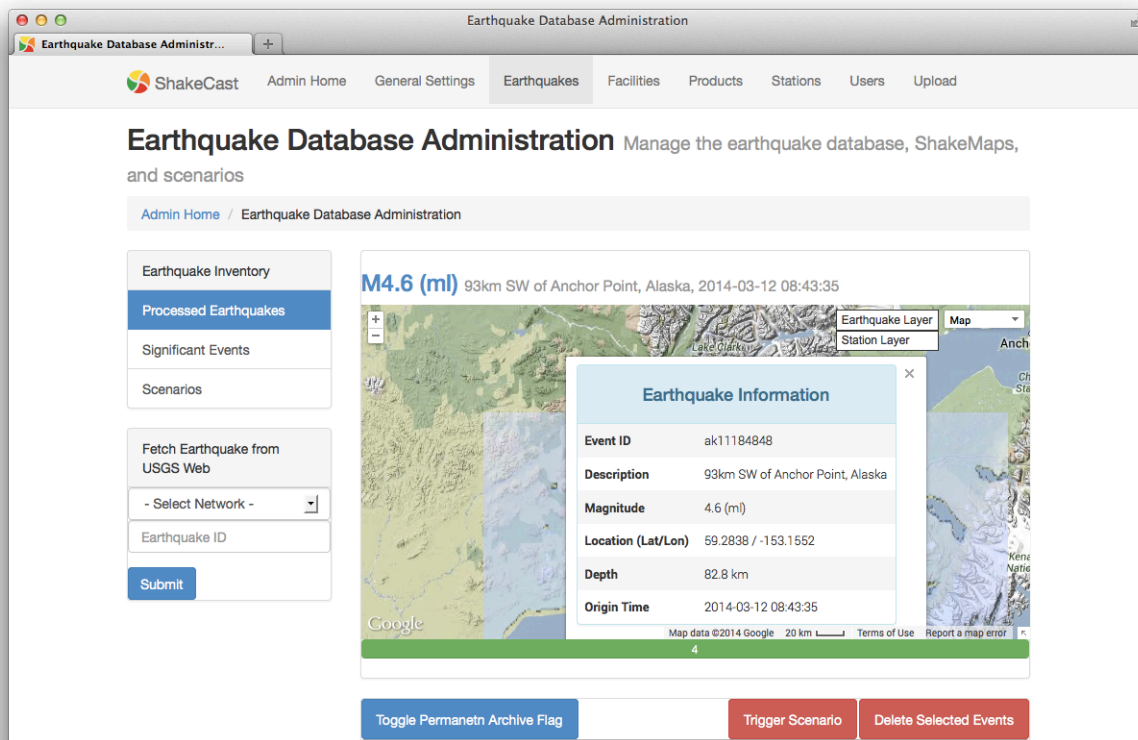


Figure 15. ShakeCast web interface for earthquake database administration.

Geospatial and magnitude filters determine the list of earthquakes processed and stored inside the ShakeCast database. The geospatial filter is based on either seismic network or user-defined polygon boundaries as a pre-processor for ShakeCast. The magnitude threshold filter is also a pre-processor filter for triggering ShakeCast process. The default 3.0 magnitude can be changed from the web interface. The archiving magnitude filter is used by the earthquake maintenance cron job for archiving purpose and the default 5.0 magnitude can only be changed inside the ShakeCast configuration file from the file system. For users within seismically active regions, these filters and archiving thresholds become key for balancing the immediate access to and the volume of ShakeMap products stored.

The ShakeCast V3 earthquake maintenance cron job runs daily and automatically maintains the earthquake inventory. Earthquakes without any facility exposure and with magnitudes below the archiving magnitude will be removed from the system once they fall outside twice the active response time window (as defined by the user) from the earthquake origin time.

The three main operations on an earthquake and its associated products via the GUI are to

- Toggle an archive flag
- Trigger a scenario run
- Delete an event

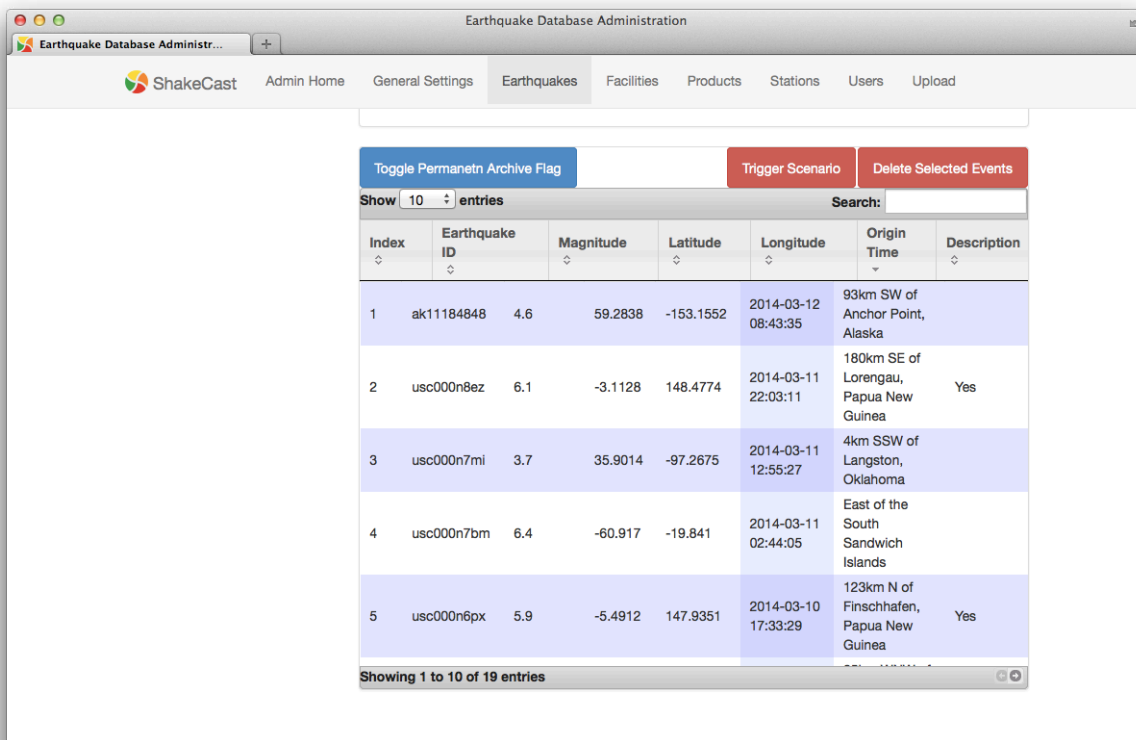


Figure 16. ShakeCast administrative web interface for earthquake database management.

The new archival flag is used to manually override the default system behavior. An earthquake will be permanently archived if the archive flag is set. All events retrieved and triggered via the web interface are treated as scenario events. To trigger a ShakeMap as an actual event, the administrator needs to perform the action from the command line to avoid accidental triggering. If the triggered ShakeMap falls outside the active response window, a “-force_run” option is required to execute the command. See Appendix C for details on triggering ShakeMaps (“shake_fetch.pl” and “scfeed_local.pl”) from the command line.

- **Processed Earthquakes:** A table of all processed actual earthquakes within twice the active response window and all archived events. Available actions are shown as clickable buttons at the top of the table. To apply the action, select the earthquake rows (multiple selections are allowed using CTRL-click or SHIFT-click) then click on the corresponding action button.
- **Significant Events:** A table of all processed actual earthquakes with a set archive flag. Available actions are the same as processed earthquakes.
- **ShakeMap Scenarios:** A table of both scenarios and actual earthquakes converted for use as scenarios. Scenario earthquakes are not subject to the archiving rules.
- **USGS ShakeMap Archive:** The USGS ShakeMap Web form reads one network ID and one event ID and retrieves the specified event as a scenario. If the event has already been processed by the system, the request will fail. The user should use the Trigger Scenario function instead. The collection of ShakeMaps for real events includes the thousands of historic events found in the ShakeMap Atlas.

- **Custom ShakeMap Scenarios:** ShakeMaps that are not available from the USGS web site can be uploaded into ShakeCast via the upload utility page. These ShakeMaps are usually custom-made based on user request or for an earthquake exercise. Most of the pre-compiled ShakeMap scenarios can be found on the ShakeCast Wiki site.

5.4 Facility Database Management

The Facility Database Management section allows the administrator to inspect facility information and to perform simple maintenance tasks. ShakeCast V3 has greatly expanded facility-related information and processing capabilities and includes:

- Basic facility and fragility information.
- Probabilistic fragility curve information.
- Supplemental attributes and facility-specific assessment methods.
- Supplemental geometric features and detailed facility information.
- Prototype facility-station association.
- Prototype predictive ground motion estimates.

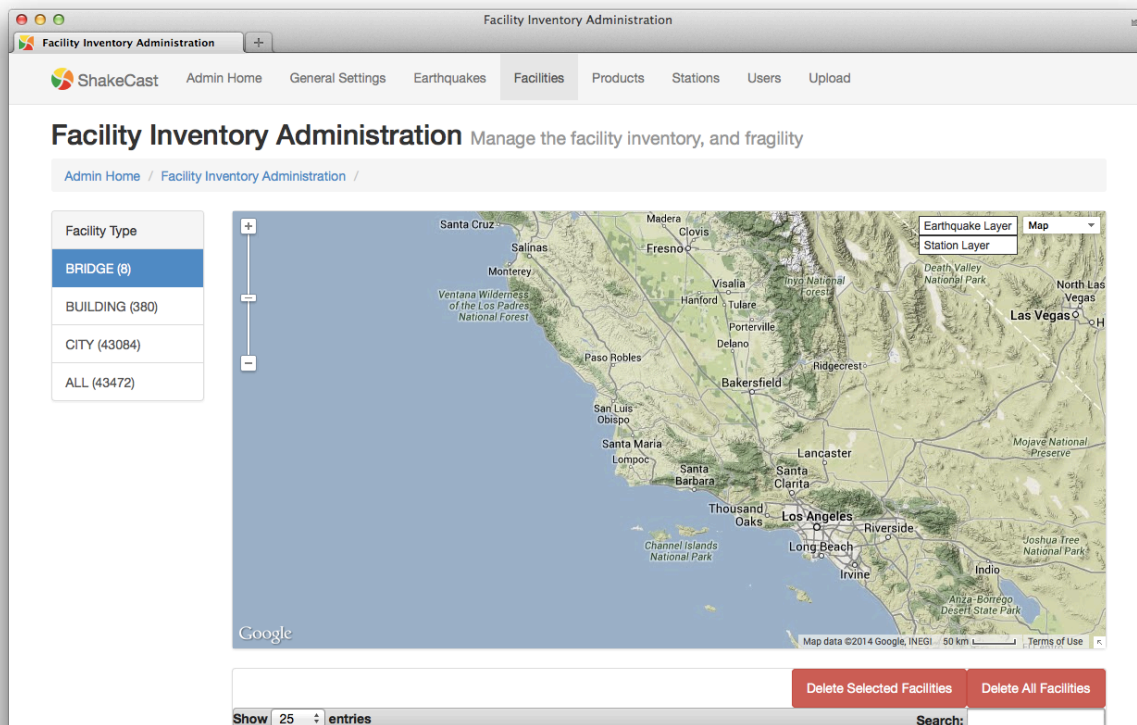
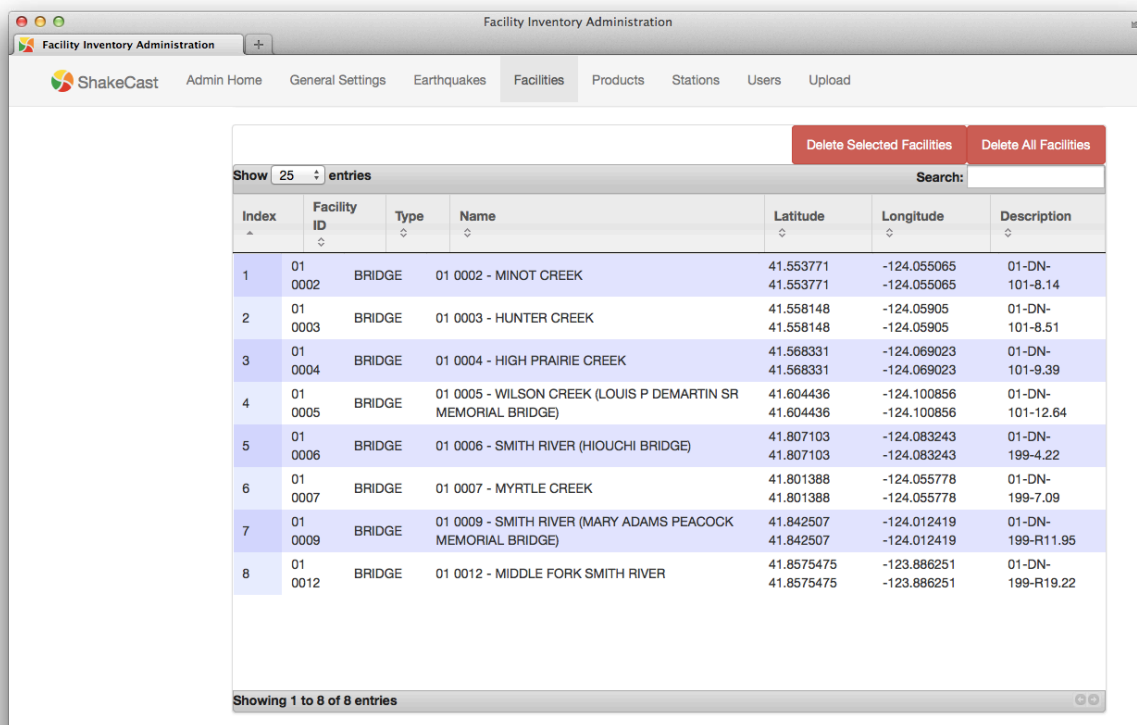


Figure 17. ShakeCast administrative web interface for facility inventory management.

The two main actions for an earthquake via the GUI are to either (1) delete selected facilities or (2) purge facilities for the selected facility type. Due to the increasing complexity of facility-related

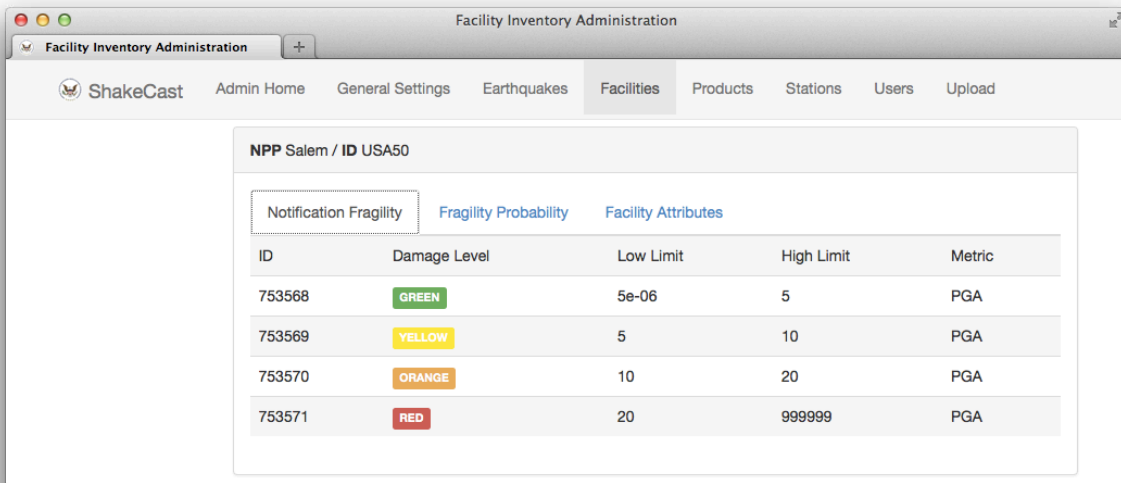
information, direct editing of facility information stored inside the ShakeCast database is disabled via the web interface. ShakeCast operators will be required to maintain their facility inventory in external data systems and produce ShakeCast compatible XML or CSV formatted files as their facility inventory and data changes over time. The XML/CSV files are then used to periodically update the facility inventory in the ShakeCast system using an upload utility page. The general approach for ShakeCast database management is that a ShakeCast operator will maintain facility data (as well as user, notification, and other ShakeCast input) offline locally to avoid editing the operational system's database, and update the operational system with pre-compiled, separately maintained data.

To view information for a single a facility, select the facility in the facility table to display detailed facility information.

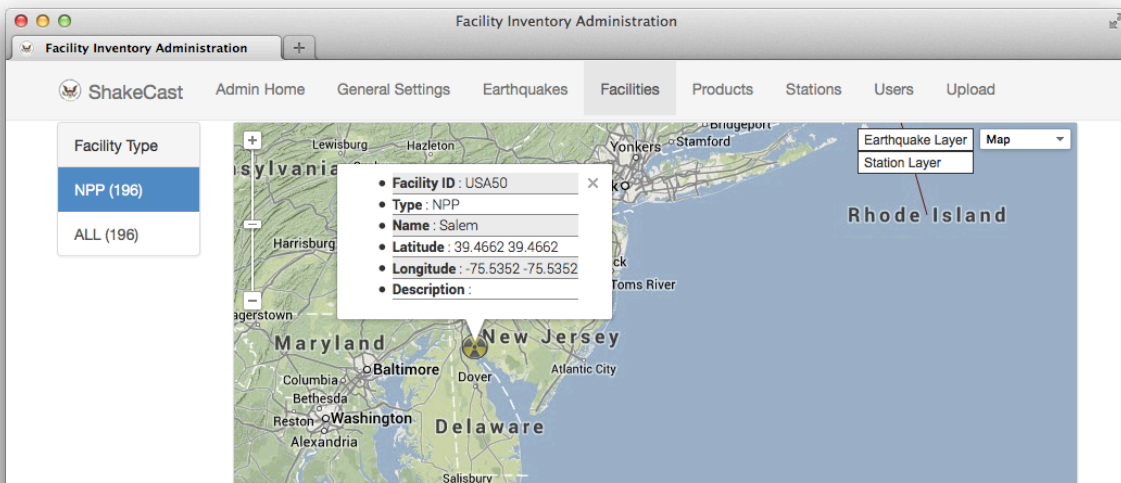


Index	Facility ID	Type	Name	Latitude	Longitude	Description
1	01 0002	BRIDGE	01 0002 - MINOT CREEK	41.553771 41.553771	-124.055065 -124.055065	01-DN-101-8.14
2	01 0003	BRIDGE	01 0003 - HUNTER CREEK	41.558148 41.558148	-124.05905 -124.05905	01-DN-101-8.51
3	01 0004	BRIDGE	01 0004 - HIGH PRAIRIE CREEK	41.568331 41.568331	-124.069023 -124.069023	01-DN-101-9.39
4	01 0005	BRIDGE	01 0005 - WILSON CREEK (LOUIS P DEMARTIN SR MEMORIAL BRIDGE)	41.604436 41.604436	-124.100856 -124.100856	01-DN-101-12.64
5	01 0006	BRIDGE	01 0006 - SMITH RIVER (HIOUCHI BRIDGE)	41.807103 41.807103	-124.083243 -124.083243	01-DN-199-4.22
6	01 0007	BRIDGE	01 0007 - MYRTLE CREEK	41.801388 41.801388	-124.055778 -124.055778	01-DN-199-7.09
7	01 0009	BRIDGE	01 0009 - SMITH RIVER (MARY ADAMS PEACOCK MEMORIAL BRIDGE)	41.842507 41.842507	-124.012419 -124.012419	01-DN-199-R11.95
8	01 0012	BRIDGE	01 0012 - MIDDLE FORK SMITH RIVER	41.8575475 41.8575475	-123.886251 -123.886251	01-DN-199-R19.22

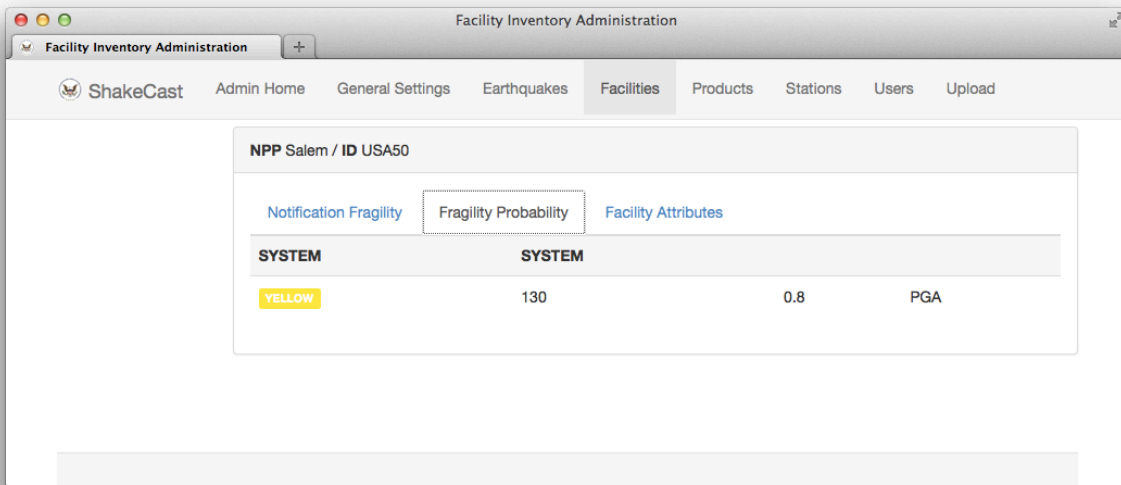
Figure 18. ShakeCast administrative web interface for facility inventory management.



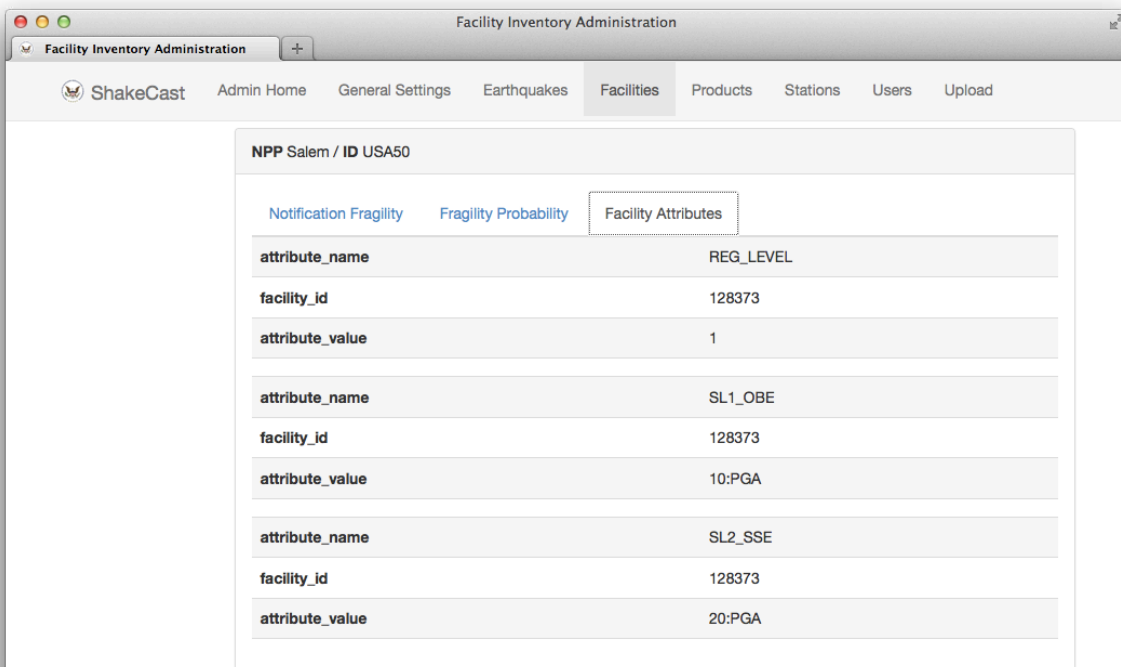
(a)



(b)



(c)



(d)

Figure 19. (a) Basic facility fragility for notification, (b) facility features, (c) fragility probability information and (d) facility attributes for a selected facility.

5.5 Earthquake Product Management

The Earthquake Product Management section allows the administrator to control and customize the products listed in the earthquake page that are presented to users, as shown in the figure below.

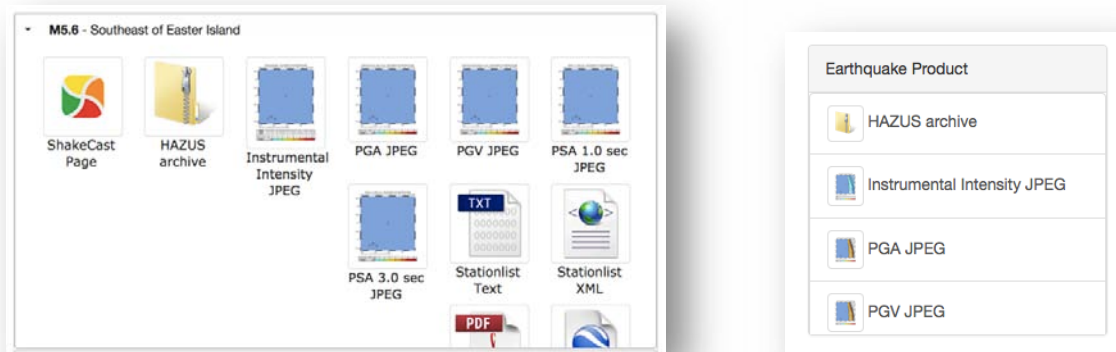


Figure 20. Earthquake product list in both the ShakeCast earthquake list and detailed page.

In ShakeCast V3, the scope of earthquake products covers not only the ShakeMap products required by ShakeCast, but also other ShakeMap products and products available from the USGS web site. Depending on the system setup, earthquake products are received via either the earthquake JSON data feed or via the PDL client. The system is pre-configured to recognize selected USGS earthquake product types including DYFI?, PAGER, ShakeMap, tectonic summary, and some local ShakeCast products. Products besides ShakeMap and ShakeCast are saved into the “eq_product” data directory. Products not registered in the ShakeCast database, such as additional DYFI? and PAGER products, geoserve, nearby_cities, and seismicity plots, are reserved for user customization in the future. Note that a product (either from USGS or locally generated) needs to be registered in ShakeCast before it can be included in the ShakeCast processing. These processes include both notification attachment and custom assessment procedure calls.

From the administrative interface, a product can be enabled or disabled for direct access by the end-user.

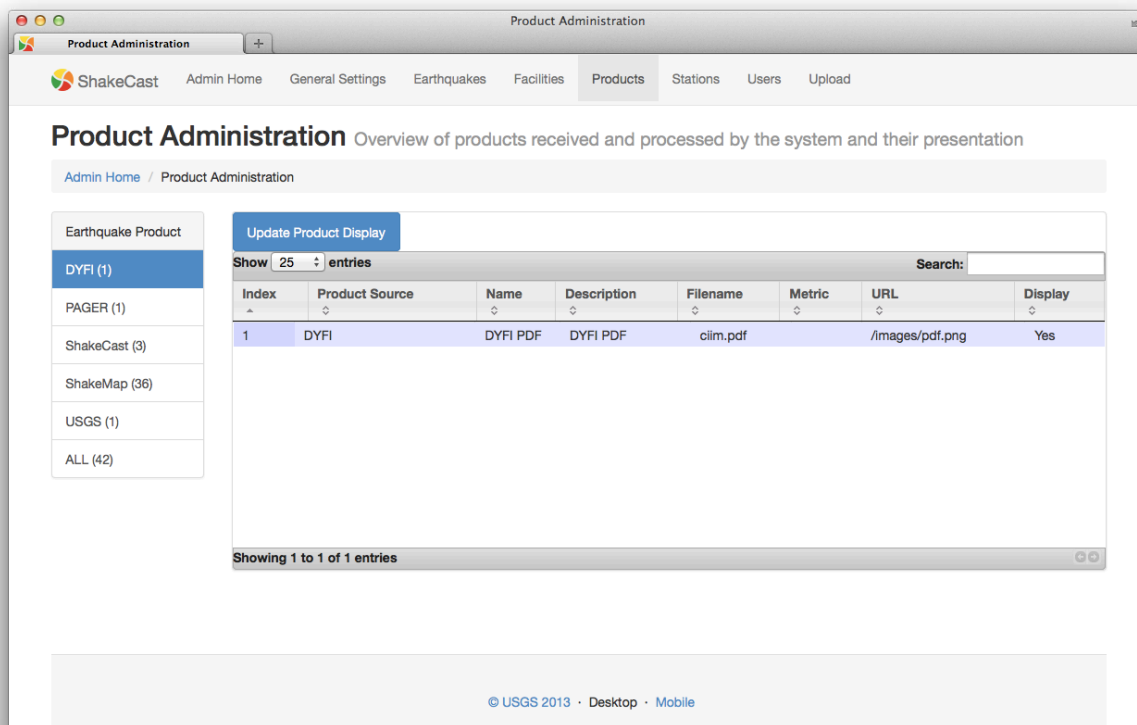


Figure 21. ShakeCast administrative web interface for earthquake product management.

5.6 Station Database Management

ShakeCast V3 comes pre-configured with a station database containing information of ~10,000 stations globally. The system will continue to add new stations to its database as it processes new station data from ShakeMap.

ShakeCast V3 includes a prototype feature that permits users to create station-facility associations to preselect actual station recordings nearby a facility to estimate from ShakeMap. The source of ground shaking data is normally based on ShakeMap input station files, but can also be provided by the user via an import program. After an earthquake, the ground shaking estimates at the site of facilities will be based on observations at station-associated facilities, or with ShakeMap estimates initially and then replaced with actual station recordings as they then become available via subsequent ShakeMap revisions or user import.

Users interested in adapting the function should be aware of potential issues pertaining to availability and quality control of strong motion data. The ShakeMap process combines predictive, actual, and converted ground motions to produce the best and stable estimates. Relying solely on a single source of data may result in unreasonable facility shaking assessment if the associated station is not properly maintained or if the recorded data is not processed correctly (e.g., clipped and non-seismic data). Also ShakeMap does not enforce the naming convention of input station data, so it is possible that the ShakeCast station database contains duplicate entries of the same station. The baseline station

information will be refreshed as part of the ShakeCast update to reflect the changes to station location and instrumentation.

From the administrative interface shown below, users can inspect the station information, but the only permitted action is to remove the selected stations from the ShakeCast database.

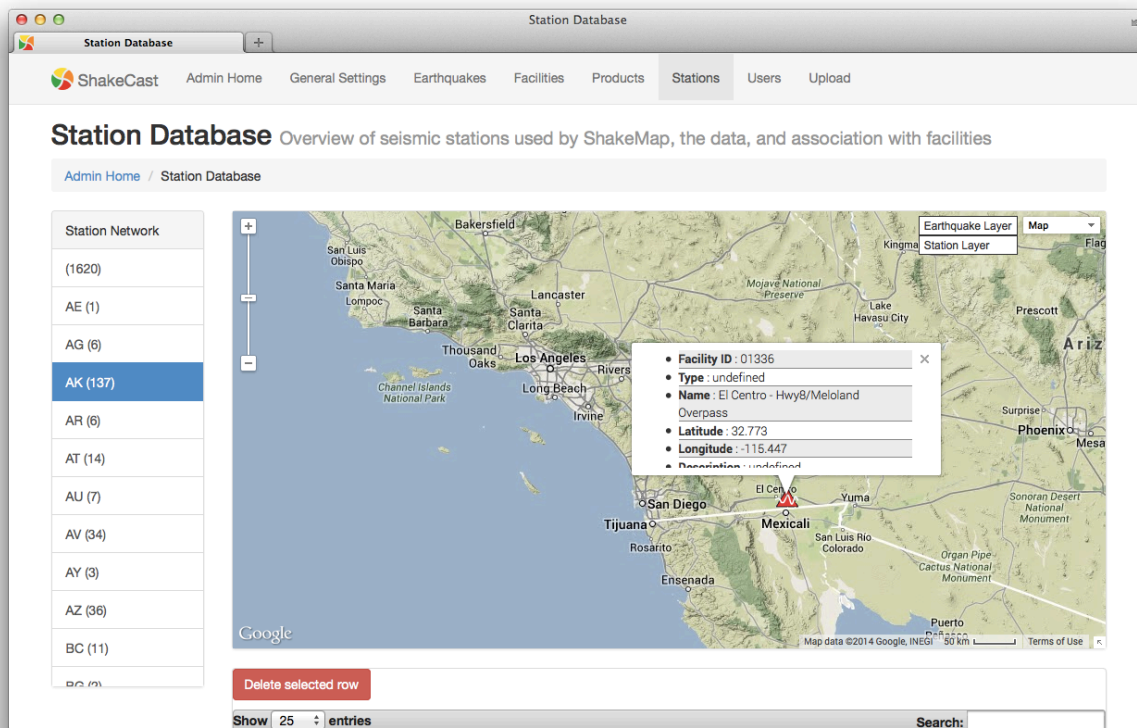


Figure 22. ShakeCast administrative web interface for station database management.

5.7 User Database Management

ShakeCast V3 defines three different user types: **ADMIN**, **USER**, and **GROUP**. The system comes pre-configured with a default administrator account “scadmin” and the administrator should change the password or remove the account before bringing the server into production.

The figure below shows the interface for user administration. The user group (**GROUP**) category operates as a universal filter for notification requests.

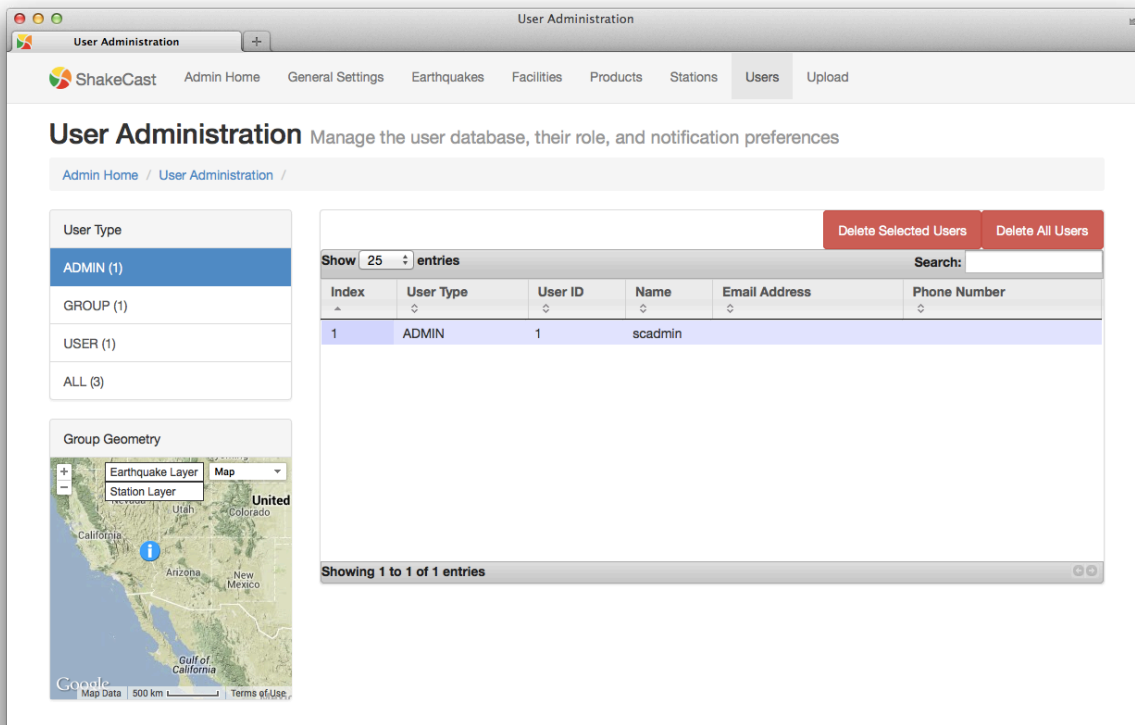


Figure 23. ShakeCast administrative web interface for user database management.

The most distinct feature of User Group is the dual purposes of the geometric polygon definition. The user-defined polygon is used to compile a list of facilities within the footprint of the polygon for notification requests. After an earthquake occurs, the same custom polygon is used as a filter to determine whether or not it should be processed by the ShakeCast system.

As a result, a user group geometry polygon is equivalent to an earthquake response region defined by the administrator. Thus the user is no longer bound to existing earthquake or ShakeMap regions defined by the seismic networks. When multiple polygons are defined, the union of the polygon footprints is the effective response region. As the default, the ShakeCast V3 system currently has one user group defined in the database with global coverage for the 40,000+ city inventory.

ShakeCast V3 only permits the **GROUP** user type to register notification requests in the ShakeCast database, both the **ADMIN** and **USER** type users need to be associated with at least one group in order to receive ShakeCast notifications. A user can be affiliated with multiple groups to receive multiple group-specific notifications. Notifications from multiple groups will not be aggregated into a single message for the user. Thus users may receive duplicate notifications if the same request is configured in separate group-specific notification requests.

Appendix A contains detailed information regarding specifications of user data, notification requests, and monitoring regions. There are two corresponding programs described in Appendix C that handle GROUP (“mange_group.pl”) and ADMIN/USER (“manage_user.pl”) data.

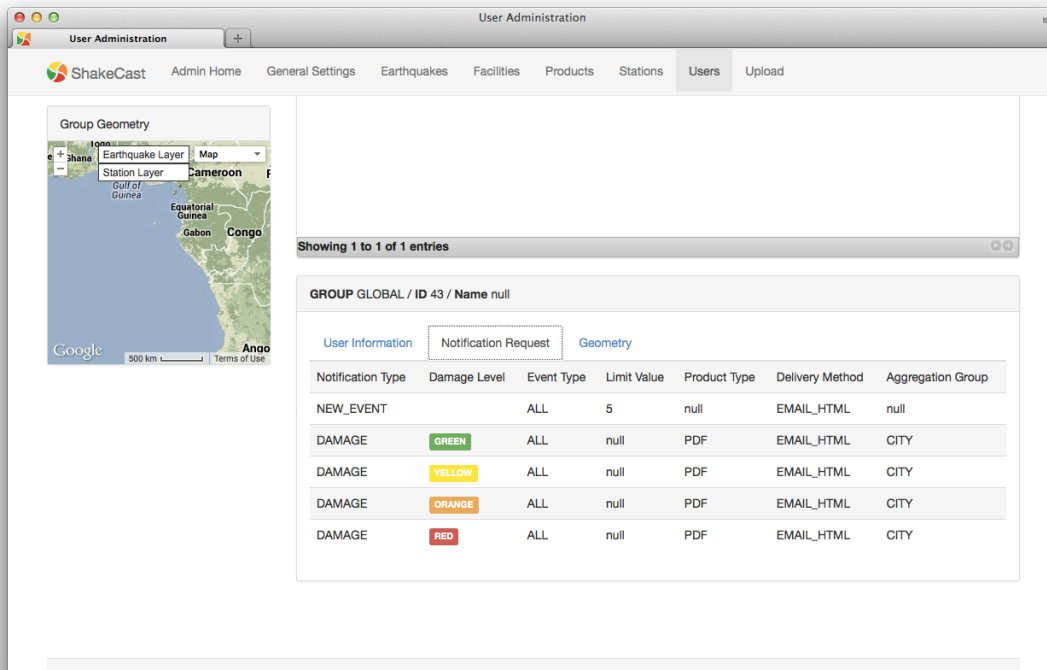


Figure 24. Notification requests for the GLOBAL group defined in the ShakeCast user database.

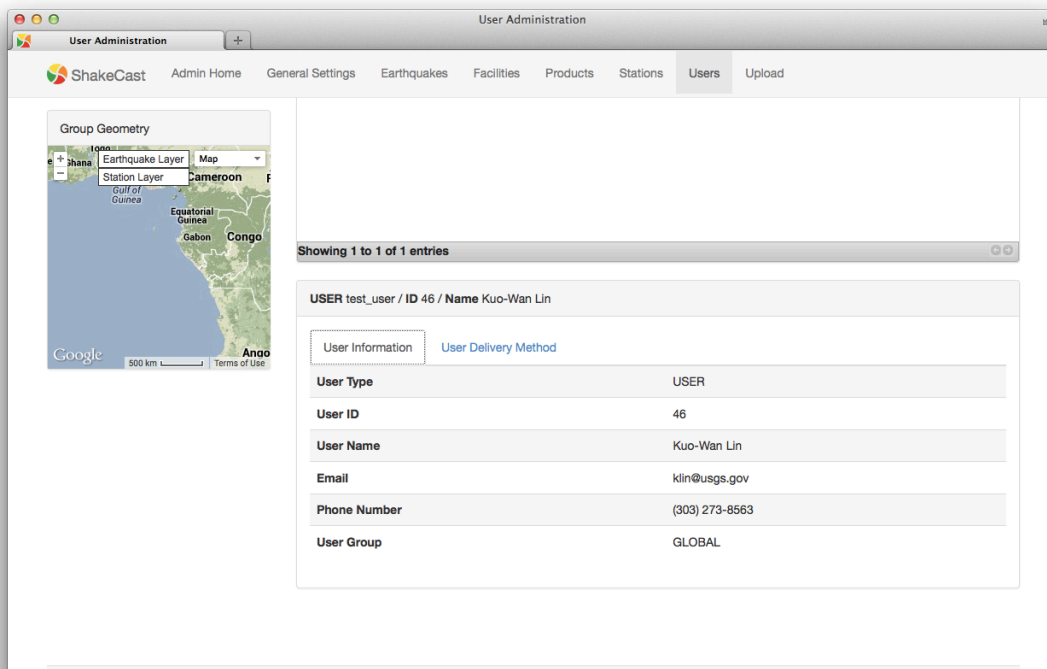


Figure 25. An example of user information and group association defined in the ShakeCast user database.

5.8 Inventory Upload Utility

ShakeCast V3 introduces a new centralized inventory upload utility page for transferring user inventory files to the ShakeCast file system. The upload page uses a drag-and-drop mechanism to provide a unified interface for all inventory types. The system allows up to five files to be uploaded simultaneously with a maximum file size of ~500MB.

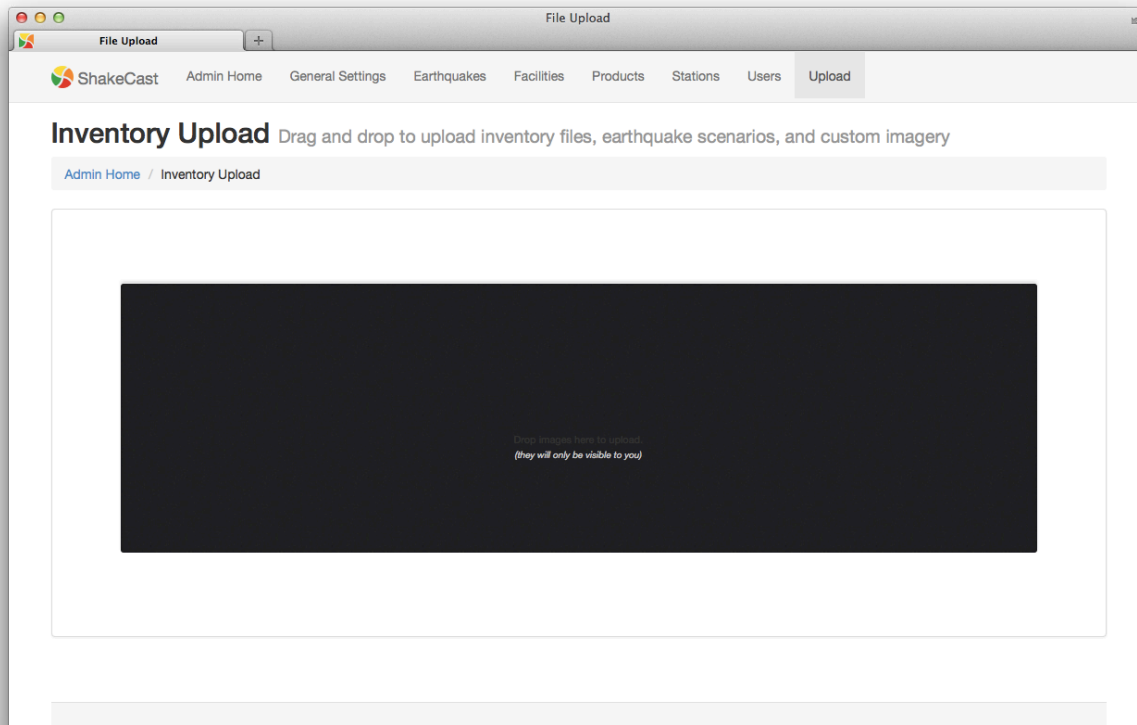


Figure 26. ShakeCast administrative interface for drag-and-drop inventory upload.

The inventory upload page displays both the status and result of individual upload attempts. All uploaded files are collected in the “tmp” directory under the ShakeCast system directory. After a successful upload, the file will be examined and the software will prompt the user with applicable choices given the nature of the file uploaded.

Recognized file types include:

- **Compressed (zip) archive:** The uploaded zip file will be uncompressed to inspect the content. ShakeMap scenario and ShakeCast test events will trigger an event processing action.
- **Configuration (conf) file:** The content of an uploaded configuration file will be examined to determine if it is a valid group configuration file. A user group processing action will be triggered if the file passes the validity checks.

- **CSV file:** The content of an uploaded CSV file will be examined to determine if it is a valid facility or user file. A user processing action will be triggered if the file is a valid user file. A CSV facility processing action will be triggered if the file is a valid facility file.
- **Image file:** gif, jpg, and png are acceptable file types. No actions will be applied. Uploaded image files will be saved as read only files into the ShakeCast image directory. This is used to upload user-specific images to overwrite the system default logo and facility icons.
- **XML file:** The content of an uploaded XML file will be examined to determine if it is a valid facility XML. An XML facility processing action will be triggered if the file passes the test.
- **All other files:** No actions will be taken if the content of an uploaded file cannot be verified. Examples of uploaded files in this category include ShakeCast patch updates and notification templates.

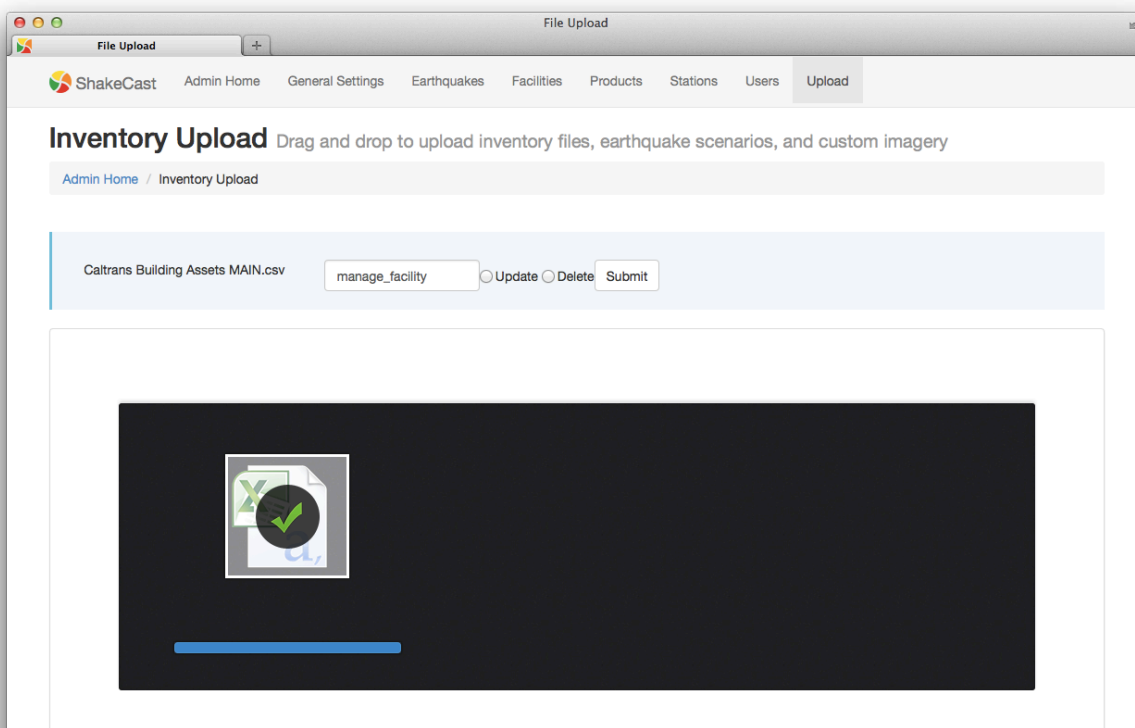


Figure 27. ShakeCast administrative interface for inventory upload showing the result of upload and the recommended user options to further process the uploaded file.

If a subsequent process is applicable to the uploaded file, a dialog form will be displayed above the drag-and-drop section. The allowed actions, insert, update and delete, will complement the inventory management described in previous sections. Results of a submitted action will be prompted as shown in the figure above. ShakeCast administrators should use the upload utility to perform common inventory maintenance tasks.

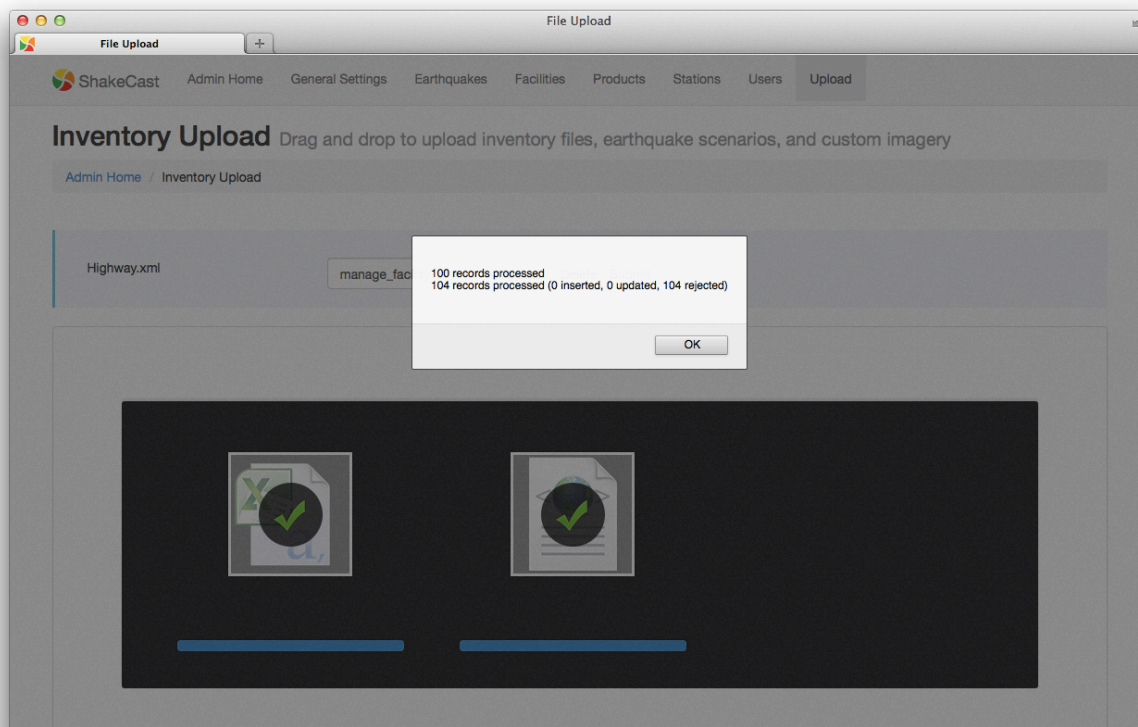


Figure 28. ShakeCast administrative interface for inventory upload showing output messages from the processing program for the uploaded file.

5.9 Cron Job

Repeated tasks are performed by a generic Cron function in ShakeCast V3. Cron Job simulates the generic cron function of a UNIX system and is platform independent. Cron jobs (crons) are saved inside the ShakeCast database managed by the ShakeCast dispatcher. The administrator does not need a separate system account with elevated credentials to create a cron job.

The scope of executable crons covers all aspects of the system and secondary computation processes to expand the core functions of the system (System Worker). Details of executable crons are described in Appendix C (task_inject), and provide the following functions:

- Compute theoretical ground motions for facilities of the specified earthquake.
- Rotate the ShakeCast log files.
- Generate log statistics plots.
- Trigger a ShakeCast heartbeat message.
- Refresh the USGS earthquake JSON feed and process new earthquakes.
- Trigger maintenance of the ShakeCast database.
- Trigger the process to compute probabilistic facility fragilities.
- Trigger the process to compute exceedance of regulatory levels.
- Take a screen shot for the selected earthquake and save the output image.

- Generate image tile overlay to be displayed on the web interface.

There are five pre-installed default cron jobs for a ShakeCast system that receive earthquake products and perform maintenance of the system. Custom configuration of cron jobs should only be performed by an experienced ShakeCast administrator.

6 References

Federal Emergency Management Agency (2006) HAZUS-MH MR2 Technical Manual: Washington, D.C., Federal Emergency Management Agency, last accessed August 22, 2008, at http://www.fema.gov/plan/prevent/hazus/hz_manuals.shtm.

Kammerer, A.M., A.R Godoy, S.Stovall, J.P.Ake, A.Altinoyollar, N.Bekiri, D. J. Wald, and K-W. Lin (2001). Developing and Implementing a Real-Time Earthquake Notification System for Nuclear Power Plant Sites using the USGS Shakecast System, *Transactions, SMiRT 21*; **6-11**, New Delhi, India, 8 pp

Appendix A ShakeCast User Data Formats

Appendix A provides specifications on the format of the data files that must be supplied to customize the system to a user's facility inventory, fragilities and their user and notification information. An example of each file is provided with the V3 ShakeCast distribution in a folder found in the ShakeCast database directory, default at "/usr/local/shakecast/sc/db". A text editor is sufficient for customizing these files, though users with substantial inventories may consider keeping a database or spreadsheet for this purpose. The suggested strategy is to maintain or export CSV files for each of the required input files described below, and then use the drag-and-drop functionality within the ShakeCast web interface to upload (or update) any of the files.

Facility Data

The scope of facility data for ShakeCast V3 covers three main categories: 1) basic facility and associated simple fragility information; 2) probabilistic fragility information; and 3) supplemental feature information. Facility data can be prepared in the format of either Comma-Separated Values (CSV) or Extensible Markup Language (2003 Excel XML export format) to be imported into the ShakeCast database. Currently there are several import scripts for processing facility data of each category and to ensure backward compatibility with V2 facility data. Specifically, the format and requirement for basic facility and associated fragility information are identical for both V2 and V3 systems. Users can migrate from a V2 system to V3 using the same facility data file.

By default CSV fields are separated by commas; field values that include commas are protected by enclosing them in quotes, but these defaults can be modified if necessary. The first record in the input file must contain column headers allowing processing scripts to interpret the rest of the records. Each header field must specify a facility field, a facility metric field, or a group field. The header fields are case-insensitive; `facility_name` and `FACILITY_NAME` are equivalent. Fields can appear in any order.

Facility Fields

The following facility names are recognized. These fields correspond to tables and columns in the ShakeCast database. Please refer to the ShakeCast Database Description for a more detailed description of the structure of the ShakeCast Database.

external_facility_id (Text(32), required always)

This field identifies the facility. It must be unique for a facility type but the same external_facility_id may be used for different types of facilities.

facility_type (Text(10), required always)

This field identifies the type of facility. It must match one of the types in the `facility_type` table. Currently defined types are: BRIDGE, CAMPUS, CITY, COUNTY, DAM, DISTRICT, ENGINEERED, INDUSTRIAL, MULTIFAM, ROAD, SINGLEFAM, STRUCTURE, TANK, TUNNEL, UNKNOWN, and HAZUS building types. Refer the HAZUS Damage Level document for the 128 HAZUS building types and code era.

facility_name (Text(128), required for insert/replace)

The value of this field is what the user sees.

short_name (Text(10), optional)

The value of this field is used by ShakeCast when a shorter version of the name is needed due to space limitations in the output.

description (Text(255), optional)

You can use this field to include a short description of the facility.

lat (Float, required for insert/replace)

Specifies the latitude of the facility in degrees and fractional degrees.

lon (Float, required for insert/replace)

Specifies the longitude of the facility in degrees and fractional degrees.

Frangility Fields

Each field beginning with `METRIC:` is taken to be a facility fragility specifier. The format of a fragility specifier is:

METRIC:*metric-name:damage-level*

where *metric-name* is a valid ShakeMap metric (MMI, PGV, PGA, PSA03, PSA10, or PSA30) and *damage-level* is a valid damage level (GREEN, YELLOW, ORANGE, or RED). Examples of Facility Fragility column labels are `METRIC:MMI:RED` and `metric:pga:yellow`.

The *metric-name* values are defined by the ShakeMap system, and are generally not changed. The above values are current as of summer 2013. The *damage-level* values shown above are the default values shipped with ShakeCast. These values are defined in your local ShakeCast database, and you may use the administration web interface to change those values and the color-names that refer to them.

Attribute Fields

A facility can have attributes associated with it. These attributes can be used to group and filter facilities.

Each field beginning with `ATTR:` is taken to be a facility attribute specifier. The format of a facility attribute specifier is:

ATTR:*attribute-name:attribute-value*

where *attribute-name* is a string not more than 20 characters in length.

Examples of Facility Attribute column labels are `ATTR:COUNTY` and `ATTR:Construction`.

Attribute values can be any string up to 30 characters long.

Example Facilities

Example 1 -- Point Facilities

Assume we have a file named `ca_cities.csv` containing California cities that we want to load into the ShakeCast database. The file is in CSV format and includes the name of each city and the latitude/longitude of its city center or city hall. Records in the file are of the form

```
Rancho Cucamonga,34.1233,-117.5794
Pasadena,34.1561,-118.1318
```

The file is missing two required fields, `external_facility_id` and `facility_type`. Since the city name is unique we can add a new column that is a copy of the name column and use that as the `external_facility_id`. Another column containing the value `CITY` for each row is added for the

facility_type. You can either make these changes using a spreadsheet program or with a simple script written in a text processing language like Perl.

After making these modifications the records look like

```
CITY,Rancho Cucamonga,Rancho Cucamonga,34.1233,-117.5794
CITY,Pasadena,Pasadena,34.1561,-118.1318
```

The input file also needs a header record; after adding one the input file looks like

```
FACILITY_TYPE,EXTERNAL_FACILITY_ID,FACILITY_NAME,LAT,LON
CITY,Rancho Cucamonga,Rancho Cucamonga,34.1233,-117.5794
CITY,Pasadena,Pasadena,34.1561,-118.1318
...
```

The facilities in this file can now be loaded into ShakeCast using the command

```
manage_facility.pl ca_cities.csv
```

Example 2 -- Fragility Parameters

Building on the previous example, assume a simple model where Instrumental Intensity (MMI) above 7 corresponds to a high-level alert (RED), MMI between 5 and 7 corresponds to a medium-level alert (YELLOW), and MMI below 5 corresponds to a low alert level (GREEN). The lower threshold of each range (1, 5, 7) is appended to every record in the input file and the header record is changed to reflect the added fields:

```
FACILITY_TYPE,EXTERNAL_FACILITY_ID,FACILITY_NAME,LAT,LON, \
    METRIC:MMI:GREEN,METRIC:MMI:YELLOW,METRIC:MMI:RED
CITY,Rancho Cucamonga,Rancho Cucamonga,34.1233,-117.5794,1,5,7
CITY,Pasadena,Pasadena,34.1561,-118.1318,1,5,7
...
```

Example 3 -- Multiple Attributes and Multiple Metrics

You can include multiple attributes, multiple metrics, or multiple attributes and multiple metrics for each row of an import file. For example,

```
FACILITY_TYPE,EXTERNAL_FACILITY_ID,ATTR:COUNTY, ATTR:SIZE, \
    METRIC:MMI:GREEN, METRIC:MMI:YELLOW, METRIC:MMI:RED
CITY,Rancho Cucamonga,San Bernardino,Small,1,2,6
CITY,Pasadena,os Angeles,Medium,1,2,6
```

This file would be loaded using the command

```
manage_facility.pl --update city_county.csv
```


The above example updates the existing city locations to associate them with a county attribute and a size attribute, and defines the green, yellow, and red shaking alert thresholds.

Probabilistic Facility Fragility Data

ShakeCast V3 includes a generic processor for evaluating probabilities of exceedance of individual damage states and their likelihood as a combined set. The probability density function is modeled as log-normal distribution. To enable this optional function, users need to provide the mean (ALPHA) and the spread (BETA) value of a fragility curve for each potential damage state to be evaluated.

When preparing probability fragility data for ShakeCast import, the format requirements for facility data are applied to fragility data. Specifically, the **external_facility_id** and **facility_type** fields of the fragility data file must match the entry in the facility data file if they are imported separately. It is permitted to define more than one set of fragility curves targeting different aspects of facility performance. This function is implemented with an additional facility attribute field called “**component**.” It is a user-defined field for describing facility-specific components to be modeled and evaluated for a given ground motion input. The user needs to specify the designated component of the facility for each fragility curve set. It is not required to define fragility curves for all potential damage states for a component. This usually applies to modeling secondary components or general stress indicators.

The user should be aware that the assessment of probabilistic analysis is considered as a secondary analysis. Results of the analysis will be stored on the ShakeCast system for inquiry by expert users but will not be sent out as the primary method for notification. Among defined components, the ALPHA value of the “SYSTEM” component will be translated as simple fragility information to be used as part of the basic facility information. This fragility information will be used for triggering notifications.

Facility Fields

The following facility names are recognized. These fields correspond to tables and columns in the ShakeCast database. Please refer to the ShakeCast Database Description for a more detailed description of the structure of the ShakeCast Database.

external_facility_id (Text(32), required always)

This field identifies the facility. It must match the one specified in the basic facility data file if the information is entered separately.

facility_type (Text(10), required always)

This field identifies the type of facility. It must match one of the types in the `facility_type` table. It must match the one specified in the basic facility data file if the information is entered separately.

class (Text(32), optional)

The value of this field is used by ShakeCast to categorize components.

component (Text(255), required always)

The value of this field is used by ShakeCast to specify the target component of the defined fragility curve.

Fragility Fields

Each field beginning with `METRIC:` is taken to be a facility fragility specifier. The format of a fragility specifier is:

`METRIC:metric-name:[ALPHA,BETA]:alert-level`

where *metric-name* is a valid Shakemap metric (MMI, PGV, PGA, PSA03, PSA10, or PSA30) and *alert-level* is a valid alert level (GREEN, YELLOW, ORANGE, or RED). Examples of Facility Fragility column labels are `METRIC:MMI:ALPHA:RED` and `metric:pga:alpha:yellow`.

The metric-name values are defined by the ShakeMap system, and are generally not changed. The above values are current as of summer 2013. The alert-level values shown above are the default values shipped with ShakeCast. These values are defined in your local ShakeCast database, and you may use the administration web interface to change those values and the color-names that refer to them.

Example Probabilistic Fragility Parameters

Assume a simple probability model where Instrumental Intensity (MMI) with ALPHA of 8 and BETA of 0.6 corresponds to a high-level alert (RED), MMI with ALPHA of 7 and BETA of 0.6 corresponds to a medium-level alert (YELLOW), and MMI with ALPHA of 5 and BETA of 0.6 corresponds to a low-level alert (GREEN). The input file and the header record is changed to reflect the added fields:

```
FACILITY_TYPE,EXTERNAL_FACILITY_ID, \
    METRIC:MMI:ALPHA:GREEN,METRIC:MMI:BETA:GREEN, \
    METRIC:MMI:ALPHA:YELLOW,METRIC:MMI:BETA:YELLOW, \
    METRIC:MMI:ALPHA:RED,METRIC:MMI:BETA:RED
CITY,Rancho Cucamonga,Rancho Cucamonga,34.1233,-117.5794,5,0.6,7,0.6,8,0.6
CITY,Pasadena,Pasadena,34.1561,-118.1318,5,0.6,7,0.6,8,0.6
...
```

Facility Feature Data

ShakeCast facility feature data allows users to define geometric footprints and rich content descriptions of a facility. The optional description field is specifically designed to store a custom HTML snippet to be displayed in the ShakeCast web interface. Content of this field will mask the data from the description field of basic facility information. The optional facility geometry field is designed to allow ShakeCast to take in account the extent of facility footprints when assessing ground shaking and damage state. The ShakeMap data grid is usually produced at a resolution of $\sim 2 \times 2 \text{ km}^2$, depending on the producer. This feature is directly applicable to facilities with linear feature such as roadways, aqueducts, etc. For facilities with small footprints there is no added benefits to define complex geometry other than either point or rectangular type.

When preparing feature data for ShakeCast import, the format requirements for facility data are applied to fragility data. Specifically, the **external_facility_id** and **facility_type** fields of the feature data file must match the entry in the facility data file if they are imported separately. The user needs to specify the geometry type of the facility for the defined geometry coordinate set. The user should be aware that the assessment of probabilistic analysis is considered as a secondary analysis. In order to accommodate the unique nature HTML snippet for the facility description, the facility feature data needs to be in the format of XML with the CSV field definition translated to tagged format.

Facility Fields

The following facility names are recognized. These fields correspond to tables and columns in the ShakeCast database. Please refer to the ShakeCast Database Description for a more detailed description of the structure of the ShakeCast Database.

external_facility_id (Text(32), required always)

This field identifies the facility. It must match the one specified in the basic facility data file if the information is entered separately.

facility_type (Text(10), required always)

This field identifies the type of facility. It must match one of the types in the `facility_type` table. It must match the one specified in the basic facility data file if the information is entered separately.

geom_type (Text(32), required always)

The value of this field is used by ShakeCast to handle the geometry coordinates from the **geom** field. Currently defined types are: POINT, POLYLINE, POLYGON, RECTANGLE, and CIRCLE.

geom (Text(mediumtext), required always)

The value of this field is used by ShakeCast to specify the coordinates of the facility. The format of this field is in (longitude,latitude) pairs separating by a white space. The size limit of data is $\sim 16\text{MB}$.

description (Text(mediumtext), required always)

You can use this field to include a description of the facility. The size limit of data is $\sim 16\text{MB}$.

Example Probabilistic Fragility Parameters

Assume a simple probability model where Instrumental Intensity (MMI) with ALPHA of 8 and BETA of 0.6 corresponds to a high-level alert (RED), MMI with ALPHA of 7 and BETA of 0.6 corresponds to

a medium-level alert (YELLOW), and MMI with ALPHA of 5 and BETA of 0.6 corresponds to low-level alert (GREEN). The input file and the header record is changed to reflect the added fields:

```
FACILITY_TYPE,EXTERNAL_FACILITY_ID, \
    METRIC:MMI:ALPHA:GREEN,METRIC:MMI:BETA:GREEN, \
    METRIC:MMI:ALPHA:YELLOW,METRIC:MMI:BETA:YELLOW, \
    METRIC:MMI:ALPHA:RED,METRIC:MMI:BETA:RED
CITY,Rancho Cucamonga,Rancho Cucamonga,34.1233,-117.5794,5,0.6,7,0.6,8,0.6
CITY,Pasadena,Pasadena,34.1561,-118.1318,5,0.6,7,0.6,8,0.6
...
```

User Group Data

ShakeCast GROUP is a new user classification introduced in V3, to provide area-specific processing and notifications for geographic regions defined by the users. For user notifications, membership in a GROUP is a primary channel of notifications for users with the same need of earthquake information.

In ShakeCast V3, GROUP polygon data is used as a geospatial filter for incoming earthquakes and ShakeMaps. This function enables ShakeCast users to define their own monitoring regions beyond the existing ShakeMap region boundaries.

User GROUP data is given in Apache config format. Lines beginning with '#' and empty lines will be ignored. Spaces at the beginning and the end of a line will also be ignored as well as tabulators. If you need spaces at the end or the beginning of a value you can use apostrophe ". An option line starts with its name followed by a value. An '=' sign is optional. Some possible examples:

```
user      max
user  = max
user                      max
```

If there is more than one statement with the same name, it will create an array instead of a scalar.

Each group is defined as a **block** of options. A **block** looks much like a block in the apache config format. It starts with **<blockname>** and ends with **</blockname>**. An example:

```
<CI>
POLY      35.8000 -116.4000  \
          34.0815 -114.4717  \
          32.0000 -114.3333  \
          32.0000 -120.5000  \
          34.5000 -121.2500  \
          37.2167 -118.0167  \
          36.6847 -117.7930  \
          35.8000 -116.4000
<NOTIFICATION>
          NOTIFICATION_TYPE    NEW_EVENT
          DELIVERY_METHOD      EMAIL_HTML
          EVENT_TYPE            ALL
</NOTIFICATION>
<NOTIFICATION>
          NOTIFICATION_TYPE    NEW_PROD
          DELIVERY_METHOD      EMAIL_HTML
          PRODUCT_TYPE          GRID_XML
          EVENT_TYPE            ALL
</NOTIFICATION>
</CI>
```

Each group is defined as a **block** of options. A **block** looks much like a block in the well known apache config format. It starts with **<blockname>** and ends with **</blockname>**. The above example defines the user group **CI**.

Group Tag Names

The following group tag names are recognized. These fields correspond to tables and columns in the ShakeCast database. Please refer to the ShakeCast Database Description for a more detailed description of the structure of the ShakeCast Database.

poly (float pairs, required always)

This field identifies the boundaries of the group geometry. It must contain at least three anchor points in order to define a polygon. The total number of anchor points should limit to less than 100, otherwise the administration interface may not be able to display the entire polygon during editing. The manage_profile.pl will however process the polygon definition.

notification (Text(32), optional)

One notification block represents one notification request associated with the group and applies to all facilities within the group polygon. Multiple notification blocks for a group are permitted.

facility_type (Text, optional)

One notification block represents facility types to be associated with the group and applies to all facilities of the specified types within the group polygon. Multiple type specifications must be separated by white spaces.

description (Text(255), optional)

One description block includes a simple description of the group.

Notification Tag Names

Each notification block defines one notification request. Tag names are corresponding to the field names of the table “notification_request.” Required tags for a notification block include NOTIFICATION_TYPE, DELIVERY_METHOD, and EVENT_TYPE. Valid notification types are CAN_EVENT, NEW_EVENT, UPD_EVENT, SHAKING, NEW_PROD, and DAMAGE.

can_event

This notification request is triggered when an event is cancelled by the seismic network in which the event was located and removed from the USGS web site. Require EVENT_TYPE and DELIVERY_METHOD tags.

new_event

This notification request is triggered when an event is located by a seismic network. A ShakeMap may or may not be produced for the earthquake depending on triggering criteria defined by the ShakeMap producers. Require EVENT_TYPE and DELIVERY_METHOD tags.

upd_event

This notification request is triggered when the source parameters of an event is updated with a new version by the seismic network. New versions of ShakeMaps for the event may or may not coincide with an updated event. Require EVENT_TYPE and DELIVERY_METHOD tags.

new_prod

This notification request is triggered when a specified ShakeMap product of an event is available on the USGS web site. Require EVENT_TYPE, DELIVERY_METHOD, and PRODUCT tags.

shaking

This notification request is triggered when the ground shaking parameter at the location of the facility exceeds the preset value. Require `EVENT_TYPE`, `DELIVERY_METHOD`, `METRIC`, and `LIMIT_VALUE` tags.

alert-level

This notification request is triggered when the ground shaking parameter at the location of the facility falls between the high and low values of facility fragility settings. Require `EVENT_TYPE`, `DELIVERY_METHOD`, and `ALERT_LEVEL` tags.

User Data

The scope of user data for ShakeCast V3 covers three user categories: 1) regular user; 2) group user; and 3) administrative user. Besides the additional group user type, there is little change to the requirements of user data and they can be prepared in the CSV format to be imported into the ShakeCast database.

Similar to facility CSV data, the first record of user data file must contain column headers. These headers tell `manage_user.pl` how to interpret the rest of the records. Each header field must specify a user name field and a user type field. The header fields are case-insensitive; `username` and `USERNAME` are equivalent. Fields can appear in any order.

User Fields

The following facility names are recognized. These fields correspond to tables and columns in the ShakeCast database. Please refer to the ShakeCast Database Description for a more detailed description of the structure of the ShakeCast Database.

username (Text(32), required always)

This field identifies the user. It must be unique for a user type.

user_type (Text(10), required always)

This field identifies the type of use. It must match one of the types in the `user_type` table.

Currently defined types are: ADMIN, USER, and SYSTEM.

full_name (Text(32), optional)

The value of this field is the user's full name.

email_address (Text(255), optional)

The value of this field is the user's email address for receiving communication from the ShakeCast system.

password (Text(64), optional)

The value of this field is used by ShakeCast to generate password for accessing the ShakeCast interface and the web site if password protected. Internally the password is saved inside the database using a cryptographic hash function SHA-256.

phone_number (Text(32), optional)

You can use this field to include a user's phone number.

Delivery Method Fields

Each field beginning with `DELIVERY:` is taken to be a delivery method specifier. The format of a delivery method specifier is:

DELIVERY:*delivery-method*

where *delivery-method* is a valid message format (PAGER, EMAIL_HTML, or EMAIL_TEXT). Examples of Delivery Method column labels are `DELIVERY:EMAIL_HTML` and `delivery:email_html`.

The message format values are defined by the ShakeCast system, and are generally not changed. The alert-level values shown above are the default values shipped with ShakeCast. These values are defined

in your local ShakeCast database, and you may use the administration web interface to change those values and the color-names that refer to them.

Group Fields

A user can have notification requests replicated from an existing group. Each field beginning with **GROUP:** is taken to be a group specifier. The format of a profile specifier is:

GROUP:*group-name[:goup-name...]*

where *group-name* is a valid group name.

User Fields

A user can have notification requests replicated from an existing user. Each field beginning with **USER:** is taken to be a user specifier. The format of a user specifier is:

USER: *shakecast-user*

where *shakecast-user* is a valid user id. In V3, use of this option is discouraged.

Example Users File

Assume we have a file named *test_user.csv* containing users that we want to load into the ShakeCast database. The file is in CSV format and includes the name of each user, user delivery method and group association. The input file with the header record looks like

```
USER_TYPE, USERNAME, PASSWORD, FULL_NAME, EMAIL_ADDRESS, PHONE_NUMBER  
, DELIVERY:PAGER, DELIVERY:EMAIL_HTML, GROUP:GLOBAL  
USER, test_user, sc4all, Test User, testuser@usgs.gov, (123) 456-7890,  
testuser@usgs.gov, testuser@usgs.gov, GLOBAL  
...
```

Plain-Text Product Template

ShakeCast V3 uses the same template engine (the Perl Template Toolkit) as V2 to generate plain-text products, such as csv, xml, and kml, etc. only available on the local ShakeCast system. A template also defined as a ShakeCast product can be included as an attachment in the ShakeCast notification to be delivered to the users directly. Please see the Template Manual manpage for the complete reference which goes into much greater details about the features and use of the Template Toolkit.

This section covers a brief summary of the template directives. ShakeCast specific identifiers include `exposure`, `item`, and `type`. Facility specific identifiers include `name`, `latitude`, `longitude`, `alert_level`, `MMI`, `PGA`, `PGV`, `PSA03`, `PSA10`, and `PSA30`.

GET

Evaluate and print a variable or value.

```
[% GET variable %]
[% variable %]
[% hash.key %]
[% list.n %]
[% code(args) %]
[% obj.meth(args) %]
[% "value: $var" %]
```

CALL

As per GET but without printing result (e.g. call code)

```
[% CALL variable %]
```

SET

Assign a values to variables.

```
[% SET variable = value %]      # 'SET' also optional
[% variable = other_variable
  variable = 'literal text @ $100'
  variable = "interpolated text: $var"
  list      = [ val, val, val, val, ... ]
  list      = [ val..val ]
  hash      = { var => val, var => val, ... }
%]
```

DEFAULT

Like SET above, but variables are only set if currently unset (i.e. have no true value).

```
[% DEFAULT variable = value %]
```

INSERT

Insert a file without any processing performed on the contents.

```
[% INSERT legalese.txt %]
```

INCLUDE

Process another template file or block and include the output. Variables are localised.

```
[% INCLUDE template %]
[% INCLUDE template var = val, ... %]
```

PROCESS

As INCLUDE above, but without localising variables.

```
[% PROCESS template %]
```

```
[% PROCESS template var = val, ... %]
```

WRAPPER

Process the enclosed block WRAPPER ... END block then INCLUDE the named template, passing the block output in the 'content' variable.

```
[% WRAPPER template %]
    content...
[% END %]
```

BLOCK

Define a named template block for subsequent INCLUDE, PROCESS, etc.,

```
[% BLOCK template %]
    content
[% END %]
```

FOREACH

Repeat the enclosed FOREACH ... END block for each value in the list.

```
[% FOREACH variable = [ val, val, val ] %]      # either
[% FOREACH variable = list %]                  # or
[% FOREACH list %]                             # or
    content...
    [% variable %]
[% END %]
```

WHILE

Enclosed WHILE ... END block is processed while condition is true.

```
[% WHILE condition %]
    content
[% END %]
```

IF / UNLESS / ELSIF / ELSE

Enclosed block is processed if the condition is true / false.

```
[% IF condition %]
    content
[% ELSIF condition %]
    content
[% ELSE %]
    content
[% END %]
[% UNLESS condition %]
    content
[% # ELSIF/ELSE as per IF, above %]
    content
[% END %]
```

SWITCH / CASE

Multi-way switch/case statement.

```
[% SWITCH variable %]
[% CASE val1 %]
    content
[% CASE [ val2, val3 ] %]
    content
```

```
[% CASE %]           # or [% CASE DEFAULT %]
    content
[% END %]
```

MACRO

Define a named macro.

```
[% MACRO name <directive> %]
[% MACRO name(arg1, arg2) <directive> %]
...
[% name %]
[% name(val1, val2) %]
```

FILTER

Process enclosed FILTER ... END block then pipe through a filter.

```
[% FILTER name %]           # either
[% FILTER name( params ) %] # or
[% FILTER alias = name( params ) %] # or
    content
[% END %]
```

USE

Load a "plugin" module, or any regular Perl module if LOAD_PERL option is set.

```
[% USE name %]           # either
[% USE name( params ) %] # or
[% USE var = name( params ) %] # or
...
[% name.method %]
[% var.method %]
```

PERL / RAWPERL

Evaluate enclosed blocks as Perl code (requires EVAL_PERL option to be set).

```
[% PERL %]
    # perl code goes here
    $stash->set('foo', 10);
    print "set 'foo' to ", $stash->get('foo'), "\n";
    print $context->include('footer', { var => $val });
[% END %]
[% RAWPERL %]
    # raw perl code goes here, no magic but fast.
    $output .= 'some output';
[% END %]
```

TRY / THROW / CATCH / FINAL

Exception handling.

```
[% TRY %]
    content
    [% THROW type info %]
[% CATCH type %]
    catch content
    [% error.type %] [% error.info %]
[% CATCH %] # or [% CATCH DEFAULT %]
```

```

        content
[% FINAL %]
    this block is always processed
[% END %]

```

NEXT

Jump straight to the next item in a FOREACH/WHILE loop.

```
[% NEXT %]
```

LAST

Break out of FOREACH/WHILE loop.

```
[% LAST %]
```

RETURN

Stop processing current template and return to including templates.

```
[% RETURN %]
```

STOP

Stop processing all templates and return to caller.

```
[% STOP %]
```

TAGS

Define new tag style or characters (default: [% %]).

```
[% TAGS html %]
[% TAGS <!-- --> %]
```

COMMENTS

Ignored and deleted.

```

[% # this is a comment to the end of line
    foo = 'bar'
%]
[%# placing the '#' immediately inside the directive
    tag comments out the entire directive
%]

```

Example Exposure Template

Assume we have a file named *exposure_csv.tt* containing template directives that we want to generate a local ShakeCast product “*exposure.csv*.” The template file first includes a static header is in CSV format. The main body of the template file contains a directive that loops through exposure facilities and outputs selected fields, including basic facility information, shaking estimates and damage estimate. The template file with the header record looks like

```

FACILITY_TYPE,FACILITY_ID,FACILITY_NAME,DIST,LATITUDE,LONGITUDE,\
DAMAGE_LEVEL,MMI,PGA,PGV,PSA03,PSA10,PSA30,STDPGA,SVEL
[% FOREACH exposure = shakecast.exposure %]
[%- FOREACH item = exposure.item -%]
[% exposure.type %], "[% item.external_facility_id %]",\
 "[% item.facility_name %]", [% item.DIST %], [% item.latitude %],\
 [% item.longitude %], [% item.damage_level %], [% item.MMI %],\
 [% item.PGA %], [% item.PGV %], [% item.PSA03 %], [% item.PSA10 %],\
 [% item.PSA30 %], [% item.STDPGA %], [% item.SVEL %]

```

```
[%- END -%]  
[% END %]
```

and the output exposure.csv looks like

```
FACILITY_TYPE,FACILITY_ID,FACILITY_NAME,DIST,LATITUDE,LONGITUDE,DAMAG  
E_LEVEL,MMI,PGA,PGV,PSA03,PSA10,PSA30,STDPGA,SVEL  
CITY,"101614","Warm Springs, NV (pop. 1K)",111.15,38.2,-  
116.4,GREEN,1,0.02,0.01,0.01,0,0,,784  
CITY,"100241","Caliente, NV (pop. 1.1K)",86.76,37.615,-  
114.511,GREEN,1.08,0.02,0.01,0.02,0,0,,483.25  
CITY,"100019","Alamo, NV (pop. < 1K)",32.61,37.365,-  
115.164,GREEN,3.08,0.15,0.04,0.15,0.02,0,,460.5  
...
```

Portable Document Format (PDF) Product Template

ShakeCast V3 introduces a new template engine to generate reports with flexible layouts in PDF format. Each PDF product template consists of one base PDF template and one configuration (or directive) file. Earthquake-specific PDF output will be saved into the earthquake-specific data directory under the same name as the PDF template. A template also defined as a ShakeCast product can be included as an attachment in the ShakeCast notification to be delivered to the users directly.

PDF directive file must be prepared in XML format. The PDF template engine runs as a middleware to translate directives to PDF layout commands. Thus although there are no ShakeCast-specific requirements, users need to refer to the Adobe PDF specifications regarding text, fonts, graphics, and other information needed to display it.

Each styled-content is defined as a **block** of options. A **block** looks much like a block in the well-known XML tag. It starts with **<blockname>** and ends with **</blockname>**. An example:

```
<image>
  <path>screenshot.jpg</path>
  <type>jpeg</type>
  <x>0</x>
  <y>0</y>
  <w>8.0</w>
  <h>4.0</h>
  <unit>inch</unit>
  <align>center</align>
  <valign>center</valign>
  <pad>0.1</pad>
</image>
```

defines the content and layout of an image.

The following PDF tag names are recognized. These fields correspond to specific PDF format specification.

page

Insert a new page in the PDF document. The example below inserts a new page in the PDF document and imports a DYFI pdf into the page.

```
<page>
  <pdf>
    <path>eq_product/[EVID]/*_ciim.pdf</path>
  </pdf>
</page>
```

block

Insert a block of content inside a page at the specified location. The example below paints a gray rectangle with black borders and inserts a paragraph of text inside the block.

```
<block fillcolor="lightgrey" strokecolor="black" >
  <action>rect</action>
  <style>fillstroke</style>
  <x>0.1</x>
  <y>8.3</y>
  <w>8.3</w>
  <h>0.7</h>
  <unit>inch</unit>
  <text>
    <string size="12" >These results are from an automated
system and users should consider the preliminary nature of this
information when making decisions relating to public safety.
ShakeCast results are often updated as additional or more accurate
earthquake information is reported or derived.</string>
    <x>0.15</x>
    <y>8.8</y>
    <w>8.2</w>
    <h>1.0</h>
    <lead>10</lead>
    <align>justify</align>
    <unit>inch</unit>
  </text>
</block>
```

text

Insert a text block at the specified location.

```
<text>
  <string size="22" type="Times-Bold" >Magnitude [MAG] -
[LOCSTRING]</string>
  <x>0.1</x>
  <y>9.9</y>
  <w>7.0</w>
  <h>0.5</h>
  <align>justify</align>
  <unit>inch</unit>
</text>
```

image

Insert an image from an external file at the specified location with respect to the event directory. If width (w) and height (h) are specified, the image will be resized to the specified dimensions. Acceptable image types are jpeg, tiff, png, gif, and gd.


```

<image>
  <path>screenshot.jpg</path>
  <type>jpeg</type>
  <x>0</x>
  <y>0</y>
  <w>8.0</w>
  <h>4.0</h>
  <unit>inch</unit>
  <align>center</align>
  <valign>center</valign>
  <pad>0.1</pad>
</image>

```

table

Insert a table from an external CSV file to the current page. The example below inserts a table to the specified location of the current page. New pages will be inserted if the length of the table exceeds the page height.

```

<table>
  <list>exposure.csv</list>
  <type>CITY USGS</type>
  <x>0.1</x>
  <w>8.3</w>
  <start_y>3.9</start_y>
  <next_y>10.75</next_y>
  <start_h>3.0</start_h>
  <next_h>10.0</next_h>
  <font_size>8</font_size>
  <padding>2</padding>
  <padding_right>2</padding_right>
  <background_color_even>snow</background_color_even>
  <background_color_odd>wheat</background_color_odd>
  <unit>inch</unit>
  <border>0.25</border>
  <border_color>snow</border_color>
  <field>FACILITY_TYPE,FACILITY_ID,FACILITY_NAME,DIST,DAMAGE_LEVEL
,MMI,PGA,PGV,PSA03,PSA10,PSA30,SDPGA,SVEL</field>
</table>

```

Appendix B ShakeCast XML/JSON Metadata Formats

Extensible Markup Language (known by the acronym XML) is a widely used and easily implemented method of exchanging data between disparate computer systems. The ShakeCast System receives ShakeMap information in XML from the USGS web server and uses XML to communicate all kinds of information between ShakeCast servers:

- Data about ShakeCast Servers and the ShakeCast software itself
- Data about events (earthquakes) and products (data files) available on the network
- Status information that helps the administrators of ShakeCast servers tell if their network is running smoothly

JavaScript Object Notation (JSON) is a text-based open standard designed for human-readable data. ShakeCast V3 adopts JSON as an alternative to the XML data for exchange of earthquake information. Specifically, the V3 system receives the USGS earthquake feed in the format of geographic data structures (GeoJSON) in order to retrieve selected earthquake products beyond ShakeMaps. The ShakeCast system also provides its own JSON data, primarily for the purpose of web presentations and for persistent data storage.

All locally generated XML and JSON files are stored in earthquake specific directory as cached content to be used primarily by the ShakeCast web server. This Section documents the ShakeCast XML and JSON file formats.

ShakeMap RSS Feed XML

RSS, which stands for “Really Simple Syndication” (sometimes called Rich Site Summary), has been adopted by news services, weblogs, and other online information services to send content to subscribers. After subscribing to an RSS feed, you will be notified when new content is available without having to visit the web site. The USGS ShakeMap RSS data feed contains

```
<?xml version="1.0"?>
<?xml-stylesheet href="shake_feed.xsl" type="text/xsl"
media="screen"?>
<rss xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:eq="http://earthquake.usgs.gov/rss/1.0/" version="2.0">
<channel>
<title>USGS Earthquake ShakeMaps</title>
<description>List of ShakeMaps for events in the last 30
days</description>
<link>http://earthquake.usgs.gov/</link>
<dc:publisher>U.S. Geological Survey</dc:publisher>
<pubDate>Mon, 16 Jul 2007 20:23:29 +0000</pubDate>
```

```

<item>
<title>6.7 - NEAR THE WEST COAST OF HONSHU, JAPAN</title>
<description><![CDATA[Date: Mon, 16 Jul 2007 01:13:27 GMT<br />Lat/Lon: 37.574/138.44<br />Depth: 49<br />]]></description>
<link>http://earthquake.usgs.gov/eqcenter/shakemap/global/shake/2007ewac/</link>
<pubDate>Mon, 16 Jul 2007 01:13:27 GMT</pubDate>
<geo:lat>37.574</geo:lat>
<geo:long>138.44</geo:long>
<dc:subject>6</dc:subject>
<eq:seconds>1184598989</eq:seconds>
<eq:depth>49</eq:depth>
<eq:region>global</eq:region>
<eq:shakethumb>http://earthquake.usgs.gov/eqcenter/images/thumbs/shakemap_global_2007ewac.jpg</eq:shakethumb>
</item>
</channel>
</rss>

```

Event XML

A ShakeCast Event is described by Event XML. A sample Event XML is below:

```

<event event_id="SAF_south7.8_se" event_version="1"
event_status="RELEASED" event_type="SCENARIO" event_name=""
event_location_description="SAF-southern M7.8 Scenario"
event_timestamp="2006-08-03 12:00:00"
external_event_id="SAF_south7.8_se" magnitude="7.8" lat="33.922270"
lon="-116.469670" />

```

Product XML

A ShakeCast Product is described by Product XML. A sample Product XML is shown in the following figure:

```

<product shakemap_id="SAF_south7.8_se" shakemap_version="1"
product_type="HAZUS" product_status="RELEASED" generating_server="1"
generation_timestamp="2007-02-08 16:07:03" lat_min="32.405603"
lat_max="35.455603" lon_min="-114.769670" lon_max="-119.353003" />

```

ShakeMap XML

A ShakeCast ShakeMap is described by ShakeMap XML. A sample ShakeMap XML is shown in the following figure:

```
<shakemap shakemap_id="SAF_south7.8_se" shakemap_version="1"
event_id="SAF_south7.8_se" event_version="1"
shakemap_status="RELEASED" generating_server="1" shakemap_region="ci"
generation_timestamp="2007-02-08 16:07:03" begin_timestamp="2007-02-
08 16:07:03" end_timestamp="2007-02-08 16:07:03" lat_min="32.405603"
lat_max="35.455603" lon_min="-119.353003" lon_max="-114.769670">
<metric metric_name="MMI" min_value="10.0000" max_value="9.4900" />
<metric metric_name="PGA" min_value="10.0002" max_value="9.9989" />
<metric metric_name="PGV" min_value="10.0000" max_value="99.9109" />
<metric metric_name="PSA03" min_value="10.0005" max_value="99.9687"
/>
<metric metric_name="PSA10" min_value="10.0007" max_value="99.9747"
/>
<metric metric_name="PSA30" min_value="1.7880" max_value="9.9989" />
</shakemap>
```

Exposure XML

A ShakeCast Exposure is described by Exposure XML. A sample Exposure XML is shown in the following figure:

```
<?xml version="1.0" encoding="UTF-8"?>
<exposure>
  xmlns:xlink="http://www.w3.org/1999/xlink"

  code_version="Pager 0.2.0"
  event_id="usneb6_06"
  version="1"
  timestamp="2006-10-11T16:07:03Z"
  source="us"
  status="RELEASED">

  <event
    type="ACTUAL"
    id="urn:earthquake.usgs.gov:origin:usneb6_06:1"
    magnitude="6.3"
    depth="17.1"
    latitude="-7.955000"
    longitude="110.430000"
    timestamp="2006-05-26T22:54:01GMT"
    description="JAVA, INDONESIA" />
```

```

    <shakemap
      code_version="3.1.1 GSM"
      id="urn:earthquake.usgs.gov:shakemap:usneb6_06:6"
      version="6"
      timestamp="2006-10-11T16:07:03Z"
      source="us"
      status="RELEASED" />

    <summary type="MMI" units="mmi">
      <bin label="I" value="1" range=" [.5,1.5) "
keywords="incomplete">
        <measure type="population" value="0" units="people"
source="landscan2005" />
      </bin>
      <bin label="II" value="2" range=" [1.5,2.5) "
keywords="incomplete">
        <measure type="population" value="0" units="people" />
      </bin>
      <bin label="III" value="3" range=" [2.5,3.5) "
keywords="incomplete">
        <measure type="population" value="963142"
units="people" />
      </bin>
    </summary>
  </exposure>

```

Facility Import XML

Facility data combining basic facility information, probabilistic fragility and feature data can be exported directly from Microsoft Excel using the XML Spreadsheet 2003 format to be imported into ShakeCast. A sample facility import XML is shown in the following figure:

```

<?xml version="1.0"?>
<?mso-application progid="Excel.Sheet"?>
<Workbook xmlns="urn:schemas-microsoft-com:office:spreadsheet"
  xmlns:o="urn:schemas-microsoft-com:office:office"
  xmlns:x="urn:schemas-microsoft-com:office:excel"
  xmlns:ss="urn:schemas-microsoft-com:office:spreadsheet"
  xmlns:html="http://www.w3.org/TR/REC-html40">
  <DocumentProperties xmlns="urn:schemas-microsoft-com:office:office">
    <Author>Lin, Kuo-wan</Author>
    <LastAuthor>Lin, Kuo-wan</LastAuthor>
    <Created>2013-08-30T19:13:27Z</Created>
    <Version>14.00</Version>
  </DocumentProperties>

```

```

<OfficeDocumentSettings xmlns="urn:schemas-microsoft-
com:office:office">
  <AllowPNG/>
</OfficeDocumentSettings>
<ExcelWorkbook xmlns="urn:schemas-microsoft-com:office:excel">
  <WindowHeight>7740</WindowHeight>
  <WindowWidth>19155</WindowWidth>
  <WindowTopX>120</WindowTopX>
  <WindowTopY>90</WindowTopY>
  <ProtectStructure>False</ProtectStructure>
  <ProtectWindows>False</ProtectWindows>
</ExcelWorkbook>
<Styles>
  <Style ss:ID="Default" ss:Name="Normal">
    <Alignment ss:Vertical="Bottom"/>
    <Borders/>
    <Font ss:FontName="Calibri" x:Family="Swiss" ss:Size="11"
ss:Color="#000000"/>
    <Interior/>
    <NumberFormat/>
    <Protection/>
  </Style>
  <Style ss:ID="s62">
    <Alignment ss:Vertical="Bottom" ss:WrapText="1"/>
    <Font ss:FontName="Calibri" x:Family="Swiss" ss:Size="11"
ss:Color="#000000"
    ss:Bold="1"/>
    <Interior ss:Color="#FFFFFF" ss:Pattern="Solid"/>
  </Style>
  <Style ss:ID="s63">
    <Alignment ss:Horizontal="Center" ss:Vertical="Bottom"
ss:WrapText="1"/>
    <Font ss:FontName="Calibri" x:Family="Swiss" ss:Size="11"
ss:Color="#000000"
    ss:Bold="1"/>
    <Interior ss:Color="#FFFFFF" ss:Pattern="Solid"/>
  </Style>
  <Style ss:ID="s64">
    <Alignment ss:Horizontal="Center" ss:Vertical="Bottom"
ss:WrapText="1"/>
    <Font ss:FontName="Calibri" x:Family="Swiss" ss:Size="11"
ss:Color="#000000"
    ss:Bold="1"/>
    <Interior ss:Color="#FFFFFF" ss:Pattern="Solid"/>
    <NumberFormat ss:Format="Fixed"/>
  </Style>

```

```

<Style ss:ID="s65">
  <Interior ss:Color="#FFFFFF" ss:Pattern="Solid"/>
</Style>
<Style ss:ID="s66">
  <Alignment ss:Horizontal="Center" ss:Vertical="Bottom"/>
  <Interior ss:Color="#FFFFFF" ss:Pattern="Solid"/>
</Style>
<Style ss:ID="s67">
  <Alignment ss:Horizontal="Center" ss:Vertical="Bottom"/>
  <Interior ss:Color="#FFFFFF" ss:Pattern="Solid"/>
  <NumberFormat ss:Format="Fixed"/>
</Style>
</Styles>
<Worksheet ss:Name="Sheet1">
  <Table ss:ExpandedColumnCount="19" ss:ExpandedRowCount="31921"
x:FullColumns="1"
  x:FullRows="1" ss:DefaultRowHeight="15">
    <Row ss:AutoFitHeight="0" ss:Height="47.25" ss:StyleID="s62">
      <Cell><Data ss:Type="String">EXTERNAL_FACILITY_ID</Data></Cell>
      <Cell><Data ss:Type="String">FACILITY_TYPE</Data></Cell>
      <Cell><Data ss:Type="String">COMPONENT_CLASS</Data></Cell>
      <Cell><Data ss:Type="String">COMPONENT</Data></Cell>
      <Cell><Data ss:Type="String">FACILITY_NAME</Data></Cell>
      <Cell><Data ss:Type="String">SHORT_NAME</Data></Cell>
      <Cell><Data ss:Type="String">DESCRIPTION</Data></Cell>
      <Cell><Data ss:Type="String">FEATURE:GEOM_TYPE</Data></Cell>
      <Cell><Data ss:Type="String">FEATURE:GEOM</Data></Cell>
      <Cell><Data ss:Type="String">FEATURE:DESCRIPTION</Data></Cell>
      <Cell ss:StyleID="s63"><Data
ss:Type="String">METRIC</Data></Cell>
      <Cell ss:StyleID="s63"><Data
ss:Type="String">METRIC:ALPHA:GREEN</Data></Cell>
      <Cell ss:StyleID="s63"><Data
ss:Type="String">METRIC:BETA:GREEN</Data></Cell>
      <Cell ss:StyleID="s64"><Data
ss:Type="String">METRIC:ALPHA:YELLOW</Data></Cell>
      <Cell ss:StyleID="s63"><Data
ss:Type="String">METRIC:BETA:YELLOW</Data></Cell>
      <Cell ss:StyleID="s64"><Data
ss:Type="String">METRIC:ALPHA:ORANGE</Data></Cell>
      <Cell ss:StyleID="s63"><Data
ss:Type="String">METRIC:BETA:ORANGE</Data></Cell>
      <Cell ss:StyleID="s64"><Data
ss:Type="String">METRIC:ALPHA:RED</Data></Cell>
      <Cell ss:StyleID="s63"><Data
ss:Type="String">METRIC:BETA:RED</Data></Cell>

```



```

        </tr>
        <td style="text-align: right; background-
color: rgb(153, 153, 153);">
        <span style="font-size:12px;"><span
style="font-family: arial,helvetica,sans-
serif;"><strong>Location:</strong></span><
t;/span></td>
        <td
style="background-color: rgb(153, 153, 153);">
        <span style="font-
size:12px;"><span style="font-family:
arial,helvetica,sans-serif;">0.08M N/O HANSON
LANE</span></span></td>
        </tr>
        <td style="text-align: right; background-color: rgb(204,
204, 204);">
        <span
style="font-size:12px;"><span style="font-
family: arial,helvetica,sans-
serif;"><strong>Description:</strong></span>
</span></td>
        <td
style="background-color: rgb(204, 204, 204);">
        <span style="font-
size:12px;"><span style="font-family:
arial,helvetica,sans-serif;">1-span; Prestressed concrete;
Slab; 12 deg skew; 13 m Max Span Length; NBI Class 501; Built 2001;
Improved 2001</span></span></td>
        </tr>
    </tbody>
</table>
</Data></Cell>
    <Cell ss:StyleID="s66"><Data ss:Type="String">PSA10</Data></Cell>
    <Cell ss:StyleID="s67"><Data ss:Type="Number">10</Data></Cell>
    <Cell ss:StyleID="s67"><Data ss:Type="Number">0.6</Data></Cell>
    <Cell ss:StyleID="s67"><Data
ss:Type="Number">98.901344820675007</Data></Cell>
    <Cell ss:StyleID="s67"><Data ss:Type="Number">0.6</Data></Cell>
    <Cell ss:StyleID="s67"><Data
ss:Type="Number">118.68161378481</Data></Cell>
    <Cell ss:StyleID="s67"><Data ss:Type="Number">0.6</Data></Cell>
    <Cell ss:StyleID="s67"><Data
ss:Type="Number">168.13228619514749</Data></Cell>
    <Cell ss:StyleID="s67"><Data ss:Type="Number">0.6</Data></Cell>
</Row>
<Row ss:StyleID="s65">
    <Cell><Data ss:Type="String">57C0705</Data></Cell>
    <Cell><Data ss:Type="String">BRIDGE_LC</Data></Cell>
    <Cell><Data ss:Type="String">GENERAL_DISTRESS</Data></Cell>
    <Cell><Data ss:Type="String">ABUTMENT</Data></Cell>

```

```

        <Cell ss:Index="11" ss:StyleID="s66"><Data
ss:Type="String">PSA10</Data></Cell>
        <Cell ss:StyleID="s67"><Data
ss:Type="Number">8.2100000000000009</Data></Cell>
        <Cell ss:StyleID="s67"><Data ss:Type="Number">0.6</Data></Cell>
        <Cell ss:StyleID="s67"><Data
ss:Type="Number">90.152101901050102</Data></Cell>
        <Cell ss:StyleID="s67"><Data ss:Type="Number">0.6</Data></Cell>
        <Cell ss:StyleID="s67"><Data ss:Type="String"></Data></Cell>
        <Cell ss:StyleID="s67"><Data ss:Type="String"></Data></Cell>
        <Cell ss:StyleID="s67"><Data ss:Type="String"></Data></Cell>
        <Cell ss:StyleID="s67"><Data ss:Type="String"></Data></Cell>
    </Row>
</Table>
<WorksheetOptions xmlns="urn:schemas-microsoft-com:office:excel">
    <PageSetup>
        <Header x:Margin="0.3"/>
        <Footer x:Margin="0.3"/>
        <PageMargins x:Bottom="0.75" x:Left="0.7" x:Right="0.7"
x:Top="0.75"/>
    </PageSetup>
    <Selected/>
    <Panes>
        <Pane>
            <Number>3</Number>
            <ActiveRow>1</ActiveRow>
            <RangeSelection>R2:R31921</RangeSelection>
        </Pane>
    </Panes>
    <ProtectObjects>False</ProtectObjects>
    <ProtectScenarios>False</ProtectScenarios>
</WorksheetOptions>
</Worksheet>
<Worksheet ss:Name="Sheet2">
    <Table ss:ExpandedColumnCount="1" ss:ExpandedRowCount="1"
x:FullColumns="1"
    x:FullRows="1" ss:DefaultRowHeight="15">
    </Table>
    <WorksheetOptions xmlns="urn:schemas-microsoft-com:office:excel">
        <PageSetup>
            <Header x:Margin="0.3"/>
            <Footer x:Margin="0.3"/>
            <PageMargins x:Bottom="0.75" x:Left="0.7" x:Right="0.7"
x:Top="0.75"/>
        </PageSetup>
        <ProtectObjects>False</ProtectObjects>

```

```

    <ProtectScenarios>False</ProtectScenarios>
  </WorksheetOptions>
</Worksheet>
<Worksheet ss:Name="Sheet3">
  <Table ss:ExpandedColumnCount="1" ss:ExpandedRowCount="1"
x:FullColumns="1"
  x:FullRows="1" ss:DefaultRowHeight="15">
  </Table>
  <WorksheetOptions xmlns="urn:schemas-microsoft-com:office:excel">
    <PageSetup>
      <Header x:Margin="0.3"/>
      <Footer x:Margin="0.3"/>
      <PageMargins x:Bottom="0.75" x:Left="0.7" x:Right="0.7"
x:Top="0.75"/>
    </PageSetup>
    <ProtectObjects>False</ProtectObjects>
    <ProtectScenarios>False</ProtectScenarios>
  </WorksheetOptions>
</Worksheet>
</Workbook>

```

Facility Feature Shaking XML

Facility feature shaking XML describes ground-shaking estimates within or along the footprints of facilities for the specified ShakeMap. It contains shaking estimates only for facilities with defined geometry feature. A sample facility feature shaking XML is shown below:

```

<?xml version="1.0" encoding="utf-8"?>
<kml>
  <grid_field index="1" name="LON"/>
  <grid_field index="2" name="LAT"/>
  <grid_field index="3" name="PGA"/>
  <grid_field index="4" name="SVEL"/>
  <grid_field index="5" name="PSA03"/>
  <grid_field index="6" name="MMI"/>
  <grid_field index="7" name="PGV"/>
  <grid_field index="8" name="PSA30"/>
  <grid_field index="9" name="PSA10"/>
  <facility id="27-A-a">
    <geom_shaking>-
117.676512368421,33.5527855263158,13.64,330,24.41,6.04,13.19,3.16,12.
95
-
117.677138857258,33.5540927826784,13.64,330,24.41,6.04,13.19,3.16,12.
95
-117.678030566667,33.5552422,13.64,330,24.41,6.04,13.19,3.16,12.95

```

```

-
117.679133671875,33.55618890625,13.64,330,24.41,6.04,13.19,3.16,12.95
-
117.680390588235,33.5569211764706,13.64,330,24.41,6.04,13.19,3.16,12.
95
-
117.681761858527,33.5574195542636,13.64,330,24.41,6.04,13.19,3.16,12.
95
-
117.683193220238,33.5576959464286,13.64,330,24.41,6.04,13.19,3.16,12.
95
</geom_shaking>
<geom_type>POLYLINE</geom_type>
</facility>
</kml>

```

USGS Earthquake JSON Feed

USGS earthquake JSON feed provides information of earthquakes and related products available on the USGS web site. A sample earthquake JSON feed is shown below:

```

{
  type: "FeatureCollection",
  metadata: {
    generated: 1379445250000,
    url:
      "http://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/1
      .0_day.geojson",
    title: "USGS Magnitude 1.0+ Earthquakes, Past Day",
    status: 200,
    api: "1.0.11",
    count: 101
  },
  features: [
    {
      type: "Feature",
      properties: {
        mag: 1.1,
        place: "41km SSW of North Pole, Alaska",
        time: 1379439188000,
        updated: 1379439776911,
        tz: -480,
        url:
          "http://earthquake.usgs.gov/earthquakes/eventpage/ak1080738
          1",

```

```

    detail:
      "http://earthquake.usgs.gov/earthquakes/feed/v1.0/detail/ak
      10807381.geojson",
      felt: null,
      cdi: null,
      mmi: null,
      alert: null,
      status: "AUTOMATIC",
      tsunami: null,
      sig: 19,
      net: "ak",
      code: "10807381",
      ids: ",ak10807381,",
      sources: ",ak,",
      types: ",general-link,geoserve,nearby-cities,origin,",
      nst: null,
      dmin: null,
      rms: 0.27,
      gap: null,
      magType: "Ml",
      type: "earthquake",
      title: "M 1.1 - 41km SSW of North Pole, Alaska"
    },
    geometry: {
      type: "Point",
      coordinates: [
        -147.7486,
        64.4188,
        9.7
      ]
    },
    id: "ak10807381"
  },

```

Facility Fragility Probability JSON

Facility fragility probability JSON describes ground shaking estimates within or along the footprints of facilities for the specified ShakeMap. It contains shaking estimates only for facilities with defined geometry feature. A sample facility feature shaking XML is shown below:

```

{
  "26074":
    [
      {
        "damage_level": "GREEN, YELLOW",
        "facility_id": "26074",

```

```

        "metric": "PGA",
        "prob_damage_level": "NA",
        "component": "LANDSLIDE",
        "class": "GROUND_FAILURE_HAZARD",
        "cdf": "0,0",
        "prob_distribution": "1,0,0"
    },
    {
        "damage_level": "GREEN,YELLOW",
        "facility_id": "26074",
        "metric": "PGA",
        "prob_damage_level": "NA",
        "component": "LIQUEFACTION",
        "class": "GROUND_FAILURE_HAZARD",
        "cdf": "0,0",
        "prob_distribution": "1,0,0"
    }
],
"25957":
[
    {
        "damage_level": "GREEN,YELLOW",
        "facility_id": "25957",
        "metric": "PGA",
        "prob_damage_level": "NA",
        "component": "LANDSLIDE",
        "class": "GROUND_FAILURE_HAZARD",
        "cdf": "0,0",
        "prob_distribution": "1,0,0"
    },
    {
        "damage_level": "GREEN,YELLOW",
        "facility_id": "25957",
        "metric": "PGA",
        "prob_damage_level": "NA",
        "component": "LIQUEFACTION",
        "class": "GROUND_FAILURE_HAZARD",
        "cdf": "0,0",
        "prob_distribution": "1,0,0"
    }
]
}

```

Event JSON

JSON equivalent of ShakeCast Event XML. A sample Event JSON is shown in the following figure:

```
{
  "shakemap_version": "1",
  "magnitude": "3.66",
  "event_id": "nn00423851",
  "lat": "37.5105",
  "superceded_timestamp": null,
  "shakemap_id": "nn00423851",
  "event_source_type": "",
  "seq": "33443",
  "mag_type": "Mwr",
  "event_name": "",
  "event_status": "NORMAL",
  "event_type": "ACTUAL",
  "event_version": "7",
  "initial_version": "0",
  "depth": "5.5",
  "external_event_id": "",
  "grid_id": "3746",
  "event_location_description": "32km WNW of Alamo, Nevada",
  "event_region": "nn",
  "event_timestamp": "2013-09-16 14:12:31",
  "lon": "-115.4841",
  "major_event": null,
  "receive_timestamp": "2013-09-16 15:43:25"
}
```

ShakeMap JSON

JSON equivalent of ShakeCast ShakeMap XML. A sample Event JSON is shown below:

```
{
  "magnitude": "3.51",
  "shakemap_version": "3",
  "event_id": "nn00423851",
  "lat": "37.5135",
  "superceded_timestamp": "2013-09-16 14:20:42",
  "metric": [
    {
      "shakemap_version": "3",
      "min_value": "1",
      "metric_name": "MMI",
      "shakemap_id": "nn00423851",
      "max_value": "5.33",
      "value_column_number": "3"
    }
  ]
}
```

```

},
{
  "shakemap_version": "3",
  "min_value": "0.01",
  "metric_name": "PGA",
  "shakemap_id": "nn00423851",
  "max_value": "3.11",
  "value_column_number": "1"
},
{
  "shakemap_version": "3",
  "min_value": "0",
  "metric_name": "PGV",
  "shakemap_id": "nn00423851",
  "max_value": "0.38",
  "value_column_number": "2"
},
{
  "shakemap_version": "3",
  "min_value": "0.01",
  "metric_name": "PSA03",
  "shakemap_id": "nn00423851",
  "max_value": "3.5",
  "value_column_number": "4"
},
{
  "shakemap_version": "3",
  "min_value": "0",
  "metric_name": "PSA10",
  "shakemap_id": "nn00423851",
  "max_value": "0.13",
  "value_column_number": "5"
},
{
  "shakemap_version": "3",
  "min_value": "0",
  "metric_name": "PSA30",
  "shakemap_id": "nn00423851",
  "max_value": "0",
  "value_column_number": "6"
},
{
  "shakemap_version": "3",
  "min_value": "301.25",
  "metric_name": "SVEL",
  "shakemap_id": "nn00423851",

```



```

        "max_value": "1061",
        "value_column_number": "8"
    }
],
"end_timestamp": "2013-09-16 15:26:55",
"shakemap_id": "nn00423851",
"lon_max": "-114.4841",
"shakemap_region": "nn",
"begin_timestamp": "2013-09-16 15:26:55",
"seq": "33435",
"lat_min": "36.7105",
"mag_type": "ml",
"event_type": "ACTUAL",
"shakemap_status": "RELEASED",
"lon_min": "-116.4841",
"depth": "5.57",
"event_version": "1",
"generation_timestamp": "2013-09-16 15:26:55",
"event_location_description": "32km WNW of Alamo, Nevada",
"lat_max": "38.3105",
"lon": "-115.4817",
"event_timestamp": "2013-09-16 14:12:31",
"generating_server": "1",
"receive_timestamp": "2013-09-16 14:19:29"
}

```

Shaking JSON

Shaking JSON describes ground shaking estimates at facility sites for the selected earthquake. A sample Shaking JSON is shown in the following figure:

```

{
  "facility_probability": {},
  "grid": {
    "shakemap_version": "3",
    "shakemap_id": "nn00423851",
    "lon_max": "-114.4841",
    "lon_min": "-116.4841",
    "origin_lon": "-115.4841",
    "grid_id": "3749",
    "latitude_cell_count": "97",
    "origin_lat": "37.5105",
    "lat_max": "38.3105",
    "longitude_cell_count": "121",
    "lat_min": "36.7105",
    "receive_timestamp": "2013-09-17 19:34:32"
  },

```

```

"facility_shaking": {
  "171187": {
    "pgv": "0.01",
    "psa10": "0",
    "facility_id": "171187",
    "svel": "784",
    "mmi": "1",
    "psa03": "0.01",
    "psa30": "0",
    "dist": "111.15",
    "pga": "0.02",
    "grid_id": "3749"
  },
  "169854": {
    "pgv": "0.01",
    "psa10": "0",
    "facility_id": "169854",
    "svel": "483.25",
    "mmi": "1.08",
    "psa03": "0.02",
    "psa30": "0",
    "dist": "86.76",
    "pga": "0.02",
    "grid_id": "3749"
  },
  "169641": {
    "pgv": "0.04",
    "psa10": "0.02",
    "facility_id": "169641",
    "svel": "460.5",
    "mmi": "3.08",
    "psa03": "0.15",
    "psa30": "0",
    "dist": "32.61",
    "pga": "0.15",
    "grid_id": "3749"
  }
}
}

```

Damage JSON

Damage JSON describes fragility settings and damage state estimates at facility sites for the selected earthquake. A sample Damage JSON is shown in the following figure:

```

{
  "facility_probability": {},

```

```

"grid": {
  "shakemap_version": "3",
  "lon_max": "-114.4841",
  "shakemap_id": "nn00423851",
  "lon_min": "-116.4841",
  "grid_id": "3749",
  "origin_lon": "-115.4841",
  "latitude_cell_count": "97",
  "origin_lat": "37.5105",
  "lat_max": "38.3105",
  "longitude_cell_count": "121",
  "lat_min": "36.7105",
  "receive_timestamp": "2013-09-16 15:29:51"
},
"damage_summary": {
  "GREEN": 2
},
"count": 2,
"facility_attribute": null,
"facility_damage": {
  "169854": {
    "psa10": "0",
    "pgv": "0.01",
    "facility_id": "169854",
    "low_limit": "1",
    "svel": "483.25",
    "lon_max": "-114.511",
    "metric": "MMI",
    "psa03": "0.02",
    "psa30": "0",
    "lat_min": "37.615",
    "damage_level": "GREEN",
    "facility_fragility_id": "877837",
    "mmi": "1.08",
    "dist": "86.76",
    "high_limit": "5",
    "lon_min": "-114.511",
    "facility_type": "CITY",
    "facility_name": "Caliente, NV (pop. 1.1K)",
    "pga": "0.02",
    "grid_id": "3749",
    "update_timestamp": null,
    "lat_max": "37.615",
    "update_username": null
  },
  "169641": {

```

```

        "psa10": "0.02",
        "pgv": "0.04",
        "facility_id": "169641",
        "low_limit": "1",
        "svel": "460.5",
        "lon_max": "-115.164",
        "metric": "MMI",
        "psa03": "0.15",
        "psa30": "0",
        "lat_min": "37.365",
        "damage_level": "GREEN",
        "facility_fragility_id": "877198",
        "mmi": "3.08",
        "dist": "32.61",
        "high_limit": "5",
        "lon_min": "-115.164",
        "facility_type": "CITY",
        "facility_name": "Alamo, NV (pop. < 1K)",
        "pga": "0.15",
        "grid_id": "3749",
        "update_timestamp": null,
        "lat_max": "37.365",
        "update_username": null
    },
    "type": null
}

```

Station JSON

Station JSON describes stations used to generate ShakeMap. A sample Station JSON is shown in the following figure:

```

[
  {
    "source": "Southern California Seismic Network",
    "commtype": "DIG",
    "longitude": "-117.43391",
    "station_id": "7",
    "update_timestamp": "2011-01-11 10:07:08",
    "latitude": "34.55046",
    "station_network": "CI",
    "station_name": "Adelanto Receiving Station",
    "external_station_id": "ADO",
    "receive_timestamp": "2013-09-17 19:41:16"
  },
  {

```

```

    "source": "Southern California Seismic Network",
    "commtype": "DIG",
    "longitude": "-118.76699",
    "station_id": "8",
    "update_timestamp": "2011-01-11 10:07:08",
    "latitude": "34.14647",
    "station_network": "CI",
    "station_name": "Agoura",
    "external_station_id": "AGO",
    "receive_timestamp": "2013-09-17 19:41:16"
  },
  {
    "source": "Southern California Seismic Network",
    "commtype": "DIG",
    "longitude": "-118.29946",
    "station_id": "9",
    "update_timestamp": "2011-01-11 10:07:08",
    "latitude": "34.68708",
    "station_network": "CI",
    "station_name": "Antelope",
    "external_station_id": "ALP",
    "receive_timestamp": "2013-09-17 19:41:16"
  },
]

```

Station JSON

Station JSON describes stations used to generate ShakeMap. A sample Station JSON is shown below:

```

[
  {
    "source": "Southern California Seismic Network",
    "commtype": "DIG",
    "longitude": "-117.43391",
    "station_id": "7",
    "update_timestamp": "2011-01-11 10:07:08",
    "latitude": "34.55046",
    "station_network": "CI",
    "station_name": "Adelanto Receiving Station",
    "external_station_id": "ADO",
    "receive_timestamp": "2013-09-17 19:41:16"
  },
  {
    "source": "Southern California Seismic Network",
    "commtype": "DIG",
    "longitude": "-118.76699",
    "station_id": "8",

```

```

        "update_timestamp": "2011-01-11 10:07:08",
        "latitude": "34.14647",
        "station_network": "CI",
        "station_name": "Agoura",
        "external_station_id": "AGO",
        "receive_timestamp": "2013-09-17 19:41:16"
    },
    {
        "source": "Southern California Seismic Network",
        "commtype": "DIG",
        "longitude": "-118.29946",
        "station_id": "9",
        "update_timestamp": "2011-01-11 10:07:08",
        "latitude": "34.68708",
        "station_network": "CI",
        "station_name": "Antelope",
        "external_station_id": "ALP",
        "receive_timestamp": "2013-09-17 19:41:16"
    },
]

```

Product JSON

JSON equivalent of ShakeCast Product XML. A sample Product JSON is shown below:

```

{
    "shakemap_version": "3",
    "shakemap_id": "nn00423851",
    "lon_max": "-114.4841",
    "lon_min": "-116.4841",
    "update_timestamp": "2013-09-16 15:29:48",
    "product_status": "RELEASED",
    "generation_timestamp": "2013-09-16 15:26:55",
    "lat_max": "38.3105",
    "product": [
        {
            "product_file_exists": "1",
            "metric": "MMI",
            "name": "Instrumental Intensity JPEG",
            "max_value": null,
            "description": null,
            "product_type": "INTEN_JPG",
            "product_id": "111052",
            "min_value": null,
            "filename": "intensity.jpg",
            "url": null
        },
    ],
}

```

```

    {
      "product_file_exists": "1",
      "metric": "PGA",
      "name": "PGA JPEG",
      "max_value": null,
      "description": null,
      "product_type": "PGA_JPG",
      "product_id": "111059",
      "min_value": null,
      "filename": "pga.jpg",
      "url": null
    }
  ]

```

Facility JSON

Facility JSON describes facilities currently populated inside the ShakeCast database. A sample Facility JSON is shown below:

```

{
  "facility_id": "171293",
  "short_name": "01 0002",
  "model": [
    [
      {
        "damage_level": "YELLOW",
        "facility_id": "171293",
        "metric": "PSA10",
        "beta": "0.6",
        "facility_fragility_model_id": "1248",
        "update_timestamp": "2013-09-02 18:21:24",
        "component": "SUPPORT_RESTRAINER",
        "alpha": "90.1521019010501",
        "class": "SECONDARY",
        "update_username": "admin"
      },
      {
        "damage_level": "GREEN",
        "facility_id": "171293",
        "metric": "PSA10",
        "beta": "0.6",
        "facility_fragility_model_id": "1247",
        "update_timestamp": "2013-09-02 18:21:24",
        "component": "SUPPORT_RESTRAINER",
        "alpha": "9.94",
        "class": "SECONDARY",
        "update_username": "admin"
      }
    ]
  ]
}

```

```

        }
    ],
    "lon_max": "-124.055065",
    "external_facility_id": "01 0002",
    "feature": [
        {
            "update_timestamp": "2013-09-02 18:21:24",
            "geom": "41.553771,-124.055065",
            "geom_type": "POINT",
            "facility_id": "171293",
            "update_username": "admin",
            "description": "    "
        }
    ],
    "attribute": [],
    "lat_min": "41.553771",
    "fragility_model": [
        {
            "abut_bearing": null,
            "system": null,
            "key": null,
            "landslide": null,
            "abutment": null,
            "support_seal": null,
            "hinge_restrainer": null,
            "abut_seal": null,
            "support_bearing": null,
            "column": null,
            "abut_restrainer": null,
            "hinge_seal": null,
            "response": null,
            "liquefaction": null,
            "hinge_bearing": null,
            "support_seat": null,
            "abut_seat": null,
            "hinge_seat": null,
            "foundation": null,
            "support_restrainer": null
        }
    ],
    "lon_min": "-124.055065",
    "description": "01-DN-101-8.14",
    "facility_name": "BRIDGE-123456",
    "facility_type": "BRIDGE",
    "update_timestamp": null,

```



```

"lat_max": "41.553771",
"fragility": [
  {
    "damage_level": "RED",
    "low_limit": "164.79416476536",
    "facility_id": "171293",
    "facility_fragility_id": "882593",
    "metric": "PSA10",
    "high_limit": "999999",
    "update_timestamp": null,
    "update_username": null
  },
  {
    "damage_level": "GREEN",
    "low_limit": "10",
    "facility_id": "171293",
    "facility_fragility_id": "882590",
    "metric": "PSA10",
    "high_limit": "96.9377439796238",
    "update_timestamp": null,
    "update_username": null
  },
  {
    "damage_level": "YELLOW",
    "low_limit": "96.9377439796238",
    "facility_id": "171293",
    "facility_fragility_id": "882591",
    "metric": "PSA10",
    "high_limit": "116.325292775549",
    "update_timestamp": null,
    "update_username": null
  },
  {
    "damage_level": "ORANGE",
    "low_limit": "116.325292775549",
    "facility_id": "171293",
    "facility_fragility_id": "882592",
    "metric": "PSA10",
    "high_limit": "164.79416476536",
    "update_timestamp": null,
    "update_username": null
  }
],
"update_username": null,
"receive_timestamp": "2013-09-17 19:53:36"
}

```

Appendix C ShakeCast Script Utilities

A number of valuable Perl scripts are distributed with ShakeCast. This Appendix documents these utilities. For the most part, these scripts are internal to the system and normally users may need only a few of them, but users with complicated databases and needs may benefit at least an awareness of their functionality.

NAME

`facility_feature_shaking.pl` – Ground Shaking Estimation Tool for Facilities with Complex Geometry

SYNOPSIS

```
facility_feature_shaking.pl <operation> [ option ... ]
```

DESCRIPTION

The **facility_feature_shaking.pl** utility is used to generate ground-shaking estimates for facilities with complex geometry. It reads one event id and one version from the command line. Output of the script is saved in `facility_feature_shaking.xml` under the directory of the specified event.

The default grid processing routine of ShakeCast handles facilities with either point or rectangle shapes. Recognized complex shapes include circle, polyline, and polygon. The shaking information is considered as secondary and cannot be used as notification thresholds.

This script is usually invoked by ShakeCast as part of the automated process. It can be run manually or by the system workers through dispatcher tasks.

OPERATIONS

event

Specify ID of the event to process.

version

Specify Version of the event to process.

OPTIONS

--verbose

Display more detailed information about the progress of the analysis. This option may be repeated to increase detail further.

--help

Print a synopsis of program usage and invocation options

NAME

facility_fragility_stat.pl – Probabilistic Fragility Estimation Tool for Facilities with Probability Fragility Curves

SYNOPSIS

```
facility_fragility_stat.pl <operation> [ option ... ]
```

DESCRIPTION

The **facility_fragility_stat.pl** utility is used to evaluate both CDF and distribution of likelihood of damage states for each identified components of the facility. It reads one event id and one version from the command line. Full fragility curve interpretation is one of the most time consuming processes of ShakeCast. The utility saves results of the analysis in both binary form “frag_prob.hash” and in text form “frag_prob.json”. The binary data is a fully structured data storage that captures a snapshot of the final data output that can be used by subsequent processes without repeating the same computation. The text-based JSON output is designed for presentations through the web interface.

The shaking information is considered as secondary and cannot be used as notification thresholds. The probability fragility tool provides detailed evaluations to complement the basic fragility and notification processes.

This script is usually invoked by ShakeCast as part of the automated process. It can be run manually or by the system workers through dispatcher tasks.

OPERATIONS

event

Specify ID of the event to process.

version

Specify Version of the event to process.

OPTIONS

--verbose

Display more detailed information about the progress of the analysis. This option may be repeated to increase detail further.

--help

Print a synopsis of program usage and invocation options

NAME

facility_regulatory_level.pl – Tool for Evaluating Exceedance of Regulatory Levels for Nuclear Power Plants

SYNOPSIS

```
facility_regulatory_level.pl <operation> [ option ... ]
```

DESCRIPTION

The **facility_regulatory_level.pl** utility is used to evaluate exceedance of regulatory levels for nuclear power plants. Regulatory levels include SL1/OBE, SL2/SSE, and Reg. 1.166 Appendix A. It reads one event id and one version from the command line. Results of evaluations are saved in the output file “facility_regulatory_level.xml” to be used in preparing the PDF reports and to be displayed through the web interface.

The shaking information is considered as secondary and cannot be used as notification thresholds. Although the script is created for the nuclear industry as part of the project requirement, it can be modified to provide user-specific criteria as a rule-based analysis tool. The regulatory level tool provides detailed evaluations to complement the basic fragility and notification processes.

This script is usually invoked by ShakeCast as part of the automated process. It can be run manually or by the system workers through dispatcher tasks.

OPERATIONS

event

Specify ID of the event to process.

version

Specify Version of the event to process.

OPTIONS

--verbose

Display more detailed information about the progress of the analysis. This option may be repeated to increase detail further.

--help

Print a synopsis of program usage and invocation options

NAME

`gs_json.pl` – USGS Earthquake JSON Feed Parser Tool

SYNOPSIS

`gs_json.pl`

DESCRIPTION

The **`gs_json.pl`** utility is used to parse the USGS earthquake JSON feed and selected products from the web server. Currently accepted earthquake product types include ShakeMap, DYFI?, PAGER, Earthquake Location Map, Historical Moment Tensor Map, Historical Seismicity Map, Tectonic Summary, Origin. The parser follows the JSON feed of individual earthquake products to download the selected products for use by the local ShakeCast system. Origin, ShakeMap, DYFI? and PAGER will invoke a Dispatcher task to handle the downloaded products.

This script is usually invoked by ShakeCast as part of the automated process. It can be run manually or queued to be handled by the system workers.

NAME

heartbeat.pl - ShakeCast Heartbeat Generator

SYNOPSIS

heartbeat.pl

DESCRIPTION

The **heartbeat.pl** utility is used to generate a ShakeCast event XML with event type as “HEARTBEAT.” The output is injected into the ShakeCast system via **sm_inject.pl** and a copy stored in the ShakeMap data directory. This will trigger an event notification to users whom are subscribed to receiving heartbeat events.

The script reads no options from the command line. To create a customized heartbeat event, edit the script located inside the ShakeCast bin directory.

NAME

logrotate.pl - ShakeCast Log File Rotation Tool

SYNOPSIS

```
logrotate.pl [ -conf config file ]
```

DESCRIPTION

The **logrotate.pl** utility is used to generate rotating backup files of ShakeCast log files (`sc.log`, `sc_access.log`, and `sc_error.log`). Configurable parameters include `rotate-time`, `max_size`, `keep-files`, `compress`, and `status-file`. The administrator can schedule a routine run of this script for maintenance of ShakeCast log files.

The script reads one optional configuration file from the command line. The default configuration file is “`sc.conf`”.

rotate-time

Specify the time windows for keeping log entries.

max_size

Specify the size limit of log files.

keep-files

Specify the number of backup log files to retain.

compress

Specify the compression option of backup log files.

status-file

Specify the filename of process status.

OPTIONS

--conf

Specify the filename of a custom configuration file to read process parameters for `logstats.pl`.

NAME

logstats.pl - ShakeCast Chart Generator for System Statistics

SYNOPSIS

```
logstats.pl [ -conf config file ]
```

DESCRIPTION

The **logstats.pl** utility is used to process ShakeCast log files (`sc.log`, `sc_access.log`, and `sc_error.log`) specified in the system configuration file and generate a set of image files in both histogram and pie charts. The daily activity chart is the default chart displayed in the default page of the Administration Web Interface. The administrator can schedule a routine run of this script to generate new statistics charts.

The script reads one optional configuration file from the command line. The default configuration file is “`sc.conf`”.

OPTIONS

--conf

Specify the filename of a custom configuration file to read process parameters for `logstats.pl`.

NAME

manage_event.pl - ShakeCast Event Management Tool

SYNOPSIS

```
manage_event.pl [ mode ] [ option ... ] event_id [event_id2 ... ]
```

DESCRIPTION

The **manage_event.pl** utility is used to re-alert, or delete processed ShakeMap events in the ShakeCast database. It reads one or more event ids from the command line. Mode is one of `-resend` or `--delete`. `manage_event.pl` will return an error message if you do not specify a mode.

--resend

Reprocess notifications for the ShakeMaps and resend notifications to users who are on the recipient list.

--delete

Delete existing events. All information for the processed ShakeMaps will be removed from the ShakeCast database but not downloaded products in the file system.

OPTIONS

--verbose

Display more detailed information about the progress of the import. This option may be repeated to increase detail further.

--help

Print a synopsis of program usage and invocation options

NAME

manage_facility.pl - ShakeCast Facility Management Tool

SYNOPSIS

```
manage_facility.pl [ mode ] [ option ... ] file.csv [ file2.csv ... ]
```

DESCRIPTION

The **manage_facility.pl** utility is used to insert, update, or delete facility data in the ShakeCast database. It reads data from one or more CSV format files. One or more files must be given on the command line. Multiple files can have different formats. Mode is one of `--insert`, `--replace`, `--delete`, `--update`, or `--skip`. `manage_facility.pl` will operate in `replace` mode if you do not specify a mode.

--insert

New facility records are inserted. It is an error for the facility to already exist; if it does the input record is skipped.

--replace

New records are inserted. If there is an existing facility it is first deleted, along with any associated attributes and fragility levels. All required facility fields must be supplied.

--delete

Delete existing facilities. All required facility fields must be supplied.

--skip

New facility records are inserted. Records for existing facilities are skipped without generating an error. The summary report will indicate how many records were skipped.

--update

Update existing facilities. If the facility does not already exist an error is issued and the record is skipped.

In this mode the only required fields are `EXTERNAL_FACILITY_ID` and `FACILITY_TYPE`. Any group values are simply added to the existing set of attributes for the facility, unless the new value matches an existing value, in which case the group value is skipped. For metrics, any metric that appears in the input will be completely replaced.

OPTIONS

--verbose

Display more detailed information about the progress of the import. This option may be repeated to increase detail further.

--help

Print a synopsis of program usage and invocation options

--limit=*n*

Terminate the import after *n* errors in input records. Set to 0 to allow an unlimited number of errors. This limit only applies to errors encountered when processing a data record from the input file. More serious errors, such as omitting a required field, will always cause the entire input file to be skipped.

--quote=*x*

Use *x* as the quote character in the input file. The default quote character is a quote ("). This character is also used as the escape character within a quoted string.

--separator=*x*

Use *x* as the field separator character in the input file. The default separator character is a comma (,).

FILE FORMAT

manage_facility.pl reads from one or more CSV-formatted files. By default fields are separated by commas and field values that include commas are protected by enclosing them in quotes, but these defaults can be modified; see the **--quote** and **--separator** options below.

The first record in the input file must contain column headers. These headers tell **manage_facility.pl** how to interpret the rest of the records. Each header field must specify a facility field, a facility metric field, or a group field. The header fields are case-insensitive; `facility_name` and `FACILITY_NAME` are equivalent. Fields can appear in any order.

Facility Fields

The following facility names are recognized. These fields correspond to tables and columns in the ShakeCast database. Please refer to the ShakeCast Database Description for a more detailed description of the structure of the ShakeCast Database.

external_facility_id (Text(32), required always)

This field identifies the facility. It must be unique for a facility type but the same `external_facility_id` may be used for different types of facilities.

facility_type (Text(10), required always)

This field identifies the type of facility. It must match one of the types in the `facility_type` table. Currently defined types are: BRIDGE, CAMPUS, CITY, COUNTY, DAM, DISTRICT, ENGINEERED, INDUSTRIAL, MULTIFAM, ROAD, SINGLEFAM, STRUCTURE, TANK, TUNNEL, UNKNOWN, and HAZUS building types. Refer the HAZUS Damage Level document for the 128 HAZUS building types and code era.

facility_name (Text(128), required for insert/replace)

The value of this field is what the user sees.

short_name (Text(10), optional)

The value of this field is used by ShakeCast when a shorter version of the name is needed due to space limitations in the output.

description (Text(255), optional)

You can use this field to include a short description of the facility.

lat (Float, required for insert/replace)

Specifies the latitude of the facility in degrees and fractional degrees.

lon (Float, required for insert/replace)

Specifies the longitude of the facility in degrees and fractional degrees.

Fragility Fields

Each field beginning with `METRIC:` is taken to be a facility fragility specifier. The format of a fragility specifier is:

METRIC:*metric-name:damage-level*

where *metric-name* is a valid Shakemap metric (MMI, PGV, PGA, PSA03, PSA10, or PSA30) and *damage-level* is a valid damage level (GREEN, YELLOW, ORANGE, or RED). Examples of Facility fragility column labels are `METRIC:MMI:RED` and `metric:pga:yellow`.

The metric-name values are defined by the ShakeMap system, and are generally not changed. The above values are current as of summer 2007. The damage-level values shown above are the default values shipped with ShakeCast. These values are defined in your local ShakeCast database, and you may use the administration web interface to change those values and the color-names that refer to them.

Attribute Fields

A facility can have attributes associated with it. These attributes can be used to group and filter facilities.

Each field beginning with `ATTR:` is taken to be a facility attribute specifier. The format of a facility attribute specifier is:

ATTR:*attribute-name:attribute-value*

where *attribute-name* is a string not more than 20 characters in length.

Examples of Facility Attribute column labels are `ATTR:COUNTY` and `ATTR:Construction`. Attribute values can be any string up to 30 characters long.

EXAMPLES

Example 1 -- Point Facilities

Assume we have a file named *ca_cities.csv* containing California cities that we want to load into the ShakeCast database. The file is in CSV format and includes the name of each city and the latitude/longitude of its city center or city hall. Records in the file are of the form

```
Rancho Cucamonga, 34.1233, -117.5794
Pasadena, 34.1561, -118.1318
```

The file is missing two required fields, `external_facility_id` and `facility_type`. Since the city name is unique we can add a new column that is a copy of the name column and use that as the `external_facility_id`. Another column containing the value `CITY` for each row is added for the `facility_type`. You can either make these changes using a spreadsheet program or with a simple script written in a text processing language like Perl.

After making these modifications the records look like

```
CITY,Rancho Cucamonga,Rancho Cucamonga,34.1233,-117.5794
CITY,Pasadena,Pasadena,34.1561,-118.1318
```

The input file also needs a header record; after adding one the input file looks like

```
FACILITY_TYPE,EXTERNAL_FACILITY_ID,FACILITY_NAME,LAT,LON
CITY,Rancho Cucamonga,Rancho Cucamonga,34.1233,-117.5794
CITY,Pasadena,Pasadena,34.1561,-118.1318
...
```

The facilities in this file can now be loaded into ShakeCast using the command
`manage_facility.pl ca_cities.csv`

Example 2 -- Fragility Parameters

It is easy to load fragility parameters for your facilities using **manage_facility.pl**. Building on the previous example, assume a simple model where Instrumental Intensity (MMI) above 7 corresponds to high-level alert (RED), MMI between 5 and 7 corresponds to medium-level alert (YELLOW), and MMI below 5 corresponds to little a low-level (GREEN). The lower threshold of each range (1, 5, 7) is appended to every record in the input file and the header record is changed to reflect the added fields:

```
FACILITY_TYPE,EXTERNAL_FACILITY_ID,FACILITY_NAME,LAT,LON, \
    METRIC:MMI:GREEN,METRIC:MMI:YELLOW,METRIC:MMI:RED
CITY,Rancho Cucamonga,Rancho Cucamonga,34.1233,-117.5794,1,5,7
CITY,Pasadena,Pasadena,34.1561,-118.1318,1,5,7
...
```

Import this file as before. New facility data will replace existing ones.

Example 3 -- Multiple Attributes and Multiple Metrics

You can include multiple attributes, multiple metrics, or multiple attributes and multiple metrics for each row of an import file. For example,

```
FACILITY_TYPE,EXTERNAL_FACILITY_ID,ATTR:COUNTY, ATTR:SIZE, \
    METRIC:MMI:GREEN, METRIC:MMI:YELLOW, METRIC:MMI:RED
CITY,Rancho Cucamonga,San Bernardino,Small,1,2,6
CITY,Pasadena,os Angeles,Medium,1,2,6
```

This file would be loaded using the command

```
manage_facility.pl --update city_county.csv
```

The above example updates the existing city locations to associate them with a county attribute and a size attribute, and defines the green, yellow, and red alert level shaking thresholds.

NAME

manage_group.pl - ShakeCast Group Management Tool

SYNOPSIS

```
manage_group.pl [ mode ] [ option ... ] [ profile.conf ]  
[ lat,lon ... ]
```

DESCRIPTION

The **manage_group.pl** utility is used to insert, update, or delete groups in the ShakeCast database and to associate facilities within the profile boundaries with the geometric profile. It reads data from a group configuration file or lat/lon pairs of a polygon from the command line. Mode is one of `--insert`, `--delete`, `--update`, or `--poly`. `manage_group.pl` will operate in `replace` mode if you do not specify a mode.

--insert

New groups are inserted. It is an error if the group already exists; if it does the input record is skipped.

--delete

Delete existing groups. All required group fields must be supplied.

--poly

Read polygon data from the command line and output facility data within the polygon boundaries.

OPTIONS

--conf

Specify the optional profile configuration file.

--verbose

Display more detailed information about the progress of the import. This option may be repeated to increase detail further.

--help

Print a synopsis of program usage and invocation options

FILE FORMAT

manage_group.pl reads data from a file in Apache config format. Lines begin with '#' and empty lines will be ignored. Spaces at the beginning and the end of a line will also be ignored as well as tabulators. If you need spaces at the end or the beginning of a value you can use apostrophe ". An option line starts with its name followed by a value. An '=' sign is optional. Some possible examples:

```
user      max  
user    = max  
user                max
```

If there is more than one statement with the same name, it will create an array instead of a scalar.

Each group is defined as a **block** of options. A **block** looks much like a block in the apache config format. It starts with **<blockname>** and ends with **</blockname>**. An example:

```
<CI>
    POLY      35.8000 -116.4000  \
              34.0815 -114.4717  \
              32.0000 -114.3333  \
              32.0000 -120.5000  \
              34.5000 -121.2500  \
              37.2167 -118.0167  \
              36.6847 -117.7930  \
              35.8000 -116.4000
    <NOTIFICATION>
        NOTIFICATION_TYPE    NEW_EVENT
        DELIVERY_METHOD      EMAIL_HTML
        EVENT_TYPE            ALL
    </NOTIFICATION>
    <NOTIFICATION>
        NOTIFICATION_TYPE    NEW_PROD
        DELIVERY_METHOD      EMAIL_HTML
        PRODUCT_TYPE          GRID_XML
        EVENT_TYPE            ALL
    </NOTIFICATION>
</CI>
```

Group Tag Names

The following profile tag names are recognized. These fields correspond to tables and columns in the ShakeCast database. Please refer to the ShakeCast Database Description for a more detailed description of the structure of the ShakeCast Database.

poly (float pairs, required always)

This field identifies the boundaries of the profile geometry. It must contain at least three anchor points in order to define a polygon. The total number of anchor points should limit to less than 100, otherwise the administration interface may not be able to display the entire polygon during editing. The `manage_profile.pl` will however process the polygon definition.

notification (Text(32), optional)

One notification block represents one notification request associated with the profile and applies to all facilities within the profile polygon. Multiple notification blocks for a profile are permitted.

Notification Tag Names

Each notification block defines one notification request. Tag names are corresponding to the field names of the table “profile_notification_request.” Required tags for a notification block include NOTIFICATION_TYPE, DELIVERY_METHOD, and EVENT_TYPE. Valid notification types are CAN_EVENT, NEW_EVENT, UPD_EVENT, SHAKING, NEW_PROD, and DAMAGE.

can_event

This notification request is triggered when an event is cancelled by the seismic network region in which the event was located and the ShakeMap removed from the USGS web site. Require `EVENT_TYPE` and `DELIVERY_METHOD` tags.

new_event

This notification request is triggered when an event is located by a seismic network region and a ShakeMap becomes available on the USGS web site. Require `EVENT_TYPE` and `DELIVERY_METHOD` tags.

upd_event

This notification request is triggered when the source parameters of an event is updated with a new version by the seismic network. New versions of ShakeMaps for the event may or may not coincide with an updated event. Require `EVENT_TYPE` and `DELIVERY_METHOD` tags.

new_prod

This notification request is triggered when a specified ShakeMap product of an event is available on the USGS web site. Require `EVENT_TYPE`, `DELIVERY_METHOD`, and `PRODUCT` tags.

shaking

This notification request is triggered when the ground shaking parameter at the location of the facility exceeds the preset value. Require `EVENT_TYPE`, `DELIVERY_METHOD`, `METRIC`, and `LIMIT_VALUE` tags.

damage

This notification request is triggered when the ground shaking parameter at the location of the facility falls between the high and low values of facility fragility settings. Require `EVENT_TYPE`, `DELIVERY_METHOD`, and `DAMAGE_LEVEL` tags.

NAME

manage_user.pl - ShakeCast User Management Tool

SYNOPSIS

```
manage_user.pl [ mode ] [ option ... ] file.csv [ file2.csv ... ]
```

DESCRIPTION

The **manage_user.pl** utility is used to insert, update, or delete user data in the ShakeCast database. It reads data from one or more CSV format files. One or more files must be given on the command line. Multiple files can have different formats. Mode is one of `--insert`, `--replace`, `--delete`, `--update`, or `--skip`. `manage_user.pl` will operate in `replace` mode if you do not specify a mode.

--insert

New user records are inserted. It is an error for the user to already exist; if it does the input record is skipped.

--replace

New records are inserted. If there is an existing user it is first deleted, along with any associated delivery addresses, notification requests and profiles. All required user fields must be supplied.

--delete

Delete existing users. All required user fields must be supplied.

--skip

New user records are inserted. Records for existing users are skipped without generating an error. The summary report will indicate how many records were skipped.

--update

Update existing users. If the user does not already exist an error is issued and the record is skipped. In this mode the only required fields are `USERNAME` and `USER_TYPE`. Any delivery methods, profiles and users for cloning that appears in the input will be completely replaced.

OPTIONS

--verbose

Display more detailed information about the progress of the import. This option may be repeated to increase detail further.

--help

Print a synopsis of program usage and invocation options

--limit=*n*

Terminate the import after *n* errors in input records. Set to 0 to allow an unlimited number of errors. This limit only applies to errors encountered when processing a data record from the input file. More serious errors, such as omitting a required field, will always cause the entire input file to be skipped.

--quote=*x*

Use *x* as the quote character in the input file. The default quote character is a quote ("). This character is also used as the escape character within a quoted string.

--separator=*x*

Use *x* as the field separator character in the input file. The default separator character is a comma (,).

FILE FORMAT

manage_user.pl reads from one or more CSV-formatted files. By default fields are separated by commas and field values that include commas are protected by enclosing them in quotes, but these defaults can be modified; see the **--quote** and **--separator** options below.

The first record in the input file must contain column headers. These headers tell **manage_user.pl** how to interpret the rest of the records. Each header field must specify a user name field and a user type field. The header fields are case-insensitive; `username` and `USERNAME` are equivalent. Fields can appear in any order.

User Fields

The following facility names are recognized. These fields correspond to tables and columns in the ShakeCast database. Please refer to the ShakeCast Database Description for a more detailed description of the structure of the ShakeCast Database.

username (Text(32), required always)

This field identifies the user. It must be unique for a user type.

user_type (Text(10), required always)

This field identifies the type of use. It must match one of the types in the `user_type` table. Currently defined types are: ADMIN, USER, and SYSTEM.

full_name (Text(32), optional)

The value of this field is the user's full name.

email_address (Text(10), optional)

The value of this field is the user's email address for receiving communication from the ShakeCast system.

password (Text(10), optional)

The value of this field is used by ShakeCast to generate password for accessing the ShakeCast interface and the web site if password protected.

phone_number (Text(255), optional)

You can use this field to include a user's phone number.

Delivery Method Fields

Each field beginning with `DELIVERY:` is taken to be a delivery method specifier. The format of a delivery method specifier is:

DELIVERY:*delivery-method*

where *delivery-method* is a valid message format (PAGER, EMAIL_HTML, or EMAIL_TEXT).

Examples of Delivery Method column labels are `DELIVERY:EMAIL_HTML` and `delivery:email_html`.

The message format values are defined by the ShakeCast system, and are generally not changed. The damage-level values shown above are the default values shipped with ShakeCast. These values are

defined in your local ShakeCast database, and you may use the administration web interface to change those values and the color-names that refer to them.

Profile Fields

A user can have notification requests replicated from an existing profile. Each field beginning with `PROFILE:` is taken to be a profile specifier. The format of a profile specifier is:

PROFILE:*profile-name*

where *profile-name* is a valid profile name.

User Fields

A user can have notification requests replicated from an existing user. Each field beginning with `USER:` is taken to be a user specifier. The format of a user specifier is:

USER:*shakecast-user*

where *shakecast-user* is a valid user id.

NAME

map_tile.pl – ShakeCast Image Tile Generation Tool

SYNOPSIS

```
map_tile.pl -type map_type [ option ... ]
```

DESCRIPTION

The **map_tile.pl** utility is used to generate images tiles to be used by the mapping engine of the web interface. Initially, the generated map tiles are used by the Google Maps API and can also be used by other mapping engines, such as the OpenLayers or OpenStreetMap. It reads one map_type from the command line. The map type includes earthquake, facility, and station.

Earthquake tiles are dynamically updated as soon as a new event is processed by the ShakeCast system. Both facility and station tiles are considered semi-static. Update of these tiles can be done manually or by creating a cron job in the database to schedule generation of tiles.

OPTIONS

--map_type

Specify type of map tile to process. The type must be either “event_tile”, “facility_tile”, or “station_tile”.

--min_zoom

Specify the minimum zoom level to process. The zoom level must be between 1 and 18.

--max_zoom

Specify the maximum zoom level to process. The zoom level must be between 1 and 18.

--rebuild

Delete all existing map tiles of the specified type before generating new map tiles.

--id

Create map tiles only for facility of the selected ID.

--verbose

Display more detailed information about the progress of the import. This option may be repeated to increase detail further.

--help

Print a synopsis of program usage and invocation options

NAME

sc_pdf.pl – ShakeCast PDF Report Generation Tool

SYNOPSIS

```
sc_pdf.pl -event event_id -version [ option ... ]
```

DESCRIPTION

The **sc_pdf.pl** utility is used to generate PDF reports for the selected earthquake. It reads one event id and one version from the command line. It will loop through all defined PDF templates in the PDF template directory. If a PDF report is successfully created, it will be registered as a local product and saved into the earthquake-specific data directory.

This script is usually invoked by ShakeCast as part of the automated process. It can be run manually or queued to be handled by the system workers.

OPTIONS

--event

Specify ID of the event to process.

--version

Specify Version of the event to process.

--verbose

Display more detailed information about the progress of the import. This option may be repeated to increase detail further.

--help

Print a synopsis of program usage and invocation options

NAME

scfeed_local.pl – ShakeMap Grid/Product Injection Tool

SYNOPSIS

```
scfeed_local.pl [ -event event_id ] [ option ... ]
```

DESCRIPTION

The **scfeed_local.pl** utility is used to process downloaded ShakeMap products located in the ShakeCast data directory. It reads one event id from the command line and creates XML messages before feeding them to ShakeCast. The injection process triggers the ShakeCast process in the same manner as for a real earthquake with respect to facility damage assessment and user notifications.

The name of an unprocessed ShakeMap must match the name of the event ID. ShakeMaps can be downloaded via the USGS ShakeMap link from the ShakeCast Administration Panel or manually from other sources. It will be renamed with the version number appended to the end of the directory name after **scfeed_local.pl** processed the ShakeMap. Outputs of ShakeCast XML files will also be stored in the same directory.

The script will quit gracefully if the ShakeMap has been processed earlier by the ShakeCast system and as a result no notifications will be delivered. To reprocess a ShakeMap that already exists in the ShakeCast system, the administrator will need to either convert the ShakeMap into a test event or delete the event first. In addition to the Administration Interface, an administrator can use the **tester.pl** utility to convert a ShakeMap to a test event and the **manage_event.pl** utility to delete a ShakeMap. The ShakeCast data directory for the deleted ShakeMap also needs to be removed from the file system before starting the reprocess procedure described earlier.

OPTIONS

--event

Specify ID of the event to process.

--scenario

Treat the ShakeMap as a scenario.

--force_run

Force ShakeCast to process the ShakeMap for events that do not meet the process criteria.

--verbose

Display more detailed information about the progress of the import. This option may be repeated to increase detail further.

--help

Print a synopsis of program usage and invocation options

NAME

shake_fetch.pl – ShakeMap Retrieval/Injection Tool

SYNOPSIS

```
shake_fetch.pl -network net_id -event event_id [ option ... ]
```

DESCRIPTION

The **shake_fetch.pl** utility is used to download and process ShakeMap located on the USGS web site. It reads one network id and one event id from the command line. ShakeMap products on the USGS web site are first downloaded before invoking the `scfeed_local` process to inject the ShakeMap into ShakeCast. The injection process is exactly the same as the `scfeed_local` script.

OPTIONS

--network

Specify ID of the network to process.

--event

Specify ID of the event to process.

--scenario

Treat the ShakeMap as a scenario.

--force_run

Force ShakeCast to process the ShakeMap for events that do not meet the process criteria.

--verbose

Display more detailed information about the progress of the import. This option may be repeated to increase detail further.

--help

Print a synopsis of program usage and invocation options

NAME

template.pl – ShakeCast General Templating Tool

SYNOPSIS

```
template.pl [ option ... ] -event event_id -template template
```

DESCRIPTION

The **template.pl** utility is used to generate ShakeCast facility summary for the specified event. The script reads at least one event ID and one template file from the command line. The output file is stored in the ShakeCast data directory for the specified event.

--event=s

Specify ID of the event to process.

--template=s

Specify filename of the template used to generate ShakeCast summary. The template files are located under the ShakeCast “template/xml” directory. The system comes with two default templates. “shakecast.tt” is the template for generating “exposure.xml” and the “kml.tt” for generating Google Earth kml format XML files.

OPTIONS

--version=n

Specify version number of the event to process.

--output=s

Specify filename of the output of ShakeCast summary. The output directory is the ShakeCast data directory for the specified event.

--help

Print a synopsis of program usage and invocation options

FILE FORMAT

template.pl is based on the Perl Template Toolkit. Please see the Template Manual manpage for the complete reference which goes into much greater details about the features and use of the Template Toolkit.

This section covers a brief summary of the template directives. ShakeCast specific identifiers include *exposure*, *item*, and *type*. Facility specific identifiers include *name*, *latitude*, *longitude*, *damage_level*, *MMI*, *PGA*, *PGV*, *PSA03*, *PSA10*, and *PSA30*.

GET

Evaluate and print a variable or value.

```
[% GET variable %]
[% variable %]
[% hash.key %]
[% list.n %]
[% code(args) %]
[% obj.meth(args) %]
[% "value: $var" %]
```

CALL

As per GET but without printing result (e.g. call code)

```
[% CALL variable %]
```

SET

Assign a values to variables.

```
[% SET variable = value %]      # 'SET' also optional
[% variable = other_variable
  variable = 'literal text @ $100'
  variable = "interpolated text: $var"
  list      = [ val, val, val, val, ... ]
  list      = [ val..val ]
  hash      = { var => val, var => val, ... }

%]
```

DEFAULT

Like SET above, but variables are only set if currently unset (i.e. have no true value).

```
[% DEFAULT variable = value %]
```

INSERT

Insert a file without any processing performed on the contents.

```
[% INSERT legalese.txt %]
```

INCLUDE

Process another template file or block and include the output. Variables are localised.

```
[% INCLUDE template %]
[% INCLUDE template var = val, ... %]
```

PROCESS

As INCLUDE above, but without localising variables.

```
[% PROCESS template %]
[% PROCESS template var = val, ... %]
```

WRAPPER

Process the enclosed block WRAPPER ... END block then INCLUDE the named template, passing the block output in the 'content' variable.

```
[% WRAPPER template %]
  content...
[% END %]
```

BLOCK

Define a named template block for subsequent INCLUDE, PROCESS, etc.,

```
[% BLOCK template %]
  content
[% END %]
```

FOREACH

Repeat the enclosed FOREACH ... END block for each value in the list.

```
[% FOREACH variable = [ val, val, val ] %]    # either
[% FOREACH variable = list %]                 # or
[% FOREACH list %]                             # or
    content...
    [% variable %]
[% END %]
```

WHILE

Enclosed WHILE ... END block is processed while condition is true.

```
[% WHILE condition %]
    content
[% END %]
```

IF / UNLESS / ELSIF / ELSE

Enclosed block is processed if the condition is true / false.

```
[% IF condition %]
    content
[% ELSIF condition %]
    content
[% ELSE %]
    content
[% END %]
[% UNLESS condition %]
    content
[% # ELSIF/ELSE as per IF, above %]
    content
[% END %]
```

SWITCH / CASE

Multi-way switch/case statement.

```
[% SWITCH variable %]
[% CASE val1 %]
    content
[% CASE [ val2, val3 ] %]
    content
[% CASE %]                # or [% CASE DEFAULT %]
    content
[% END %]
```

MACRO

Define a named macro.

```
[% MACRO name <directive> %]
[% MACRO name(arg1, arg2) <directive> %]
...
[% name %]
[% name(val1, val2) %]
```

FILTER

Process enclosed FILTER ... END block then pipe through a filter.

```
[% FILTER name %]                                # either
```

```
[% FILTER name( params ) %]           # or
[% FILTER alias = name( params ) %]    # or
    content
[% END %]
```

USE

Load a "plugin" module, or any regular Perl module if `LOAD_PERL` option is set.

```
[% USE name %]                         # either
[% USE name( params ) %]               # or
[% USE var = name( params ) %]         # or
...
[% name.method %]
[% var.method %]
```

PERL / RAWPERL

Evaluate enclosed blocks as Perl code (requires `EVAL_PERL` option to be set).

```
[% PERL %]
    # perl code goes here
    $stash->set('foo', 10);
    print "set 'foo' to ", $stash->get('foo'), "\n";
    print $context->include('footer', { var => $val });
[% END %]
[% RAWPERL %]
    # raw perl code goes here, no magic but fast.
    $output .= 'some output';
[% END %]
```

TRY / THROW / CATCH / FINAL

Exception handling.

```
[% TRY %]
    content
    [% THROW type info %]
[% CATCH type %]
    catch content
    [% error.type %] [% error.info %]
[% CATCH %] # or [% CATCH DEFAULT %]
    content
[% FINAL %]
    this block is always processed
[% END %]
```

NEXT

Jump straight to the next item in a `FOREACH/WHILE` loop.

```
[% NEXT %]
```

LAST

Break out of `FOREACH/WHILE` loop.

```
[% LAST %]
```

RETURN

Stop processing current template and return to including templates.

```
[% RETURN %]
```

STOP

Stop processing all templates and return to caller.

```
[% STOP %]
```

TAGS

Define new tag style or characters (default: [% %]).

```
[% TAGS html %]
```

```
[% TAGS <!-- --> %]
```

COMMENTS

Ignored and deleted.

```
[% # this is a comment to the end of line  
    foo = 'bar'
```

```
%]
```

```
[%# placing the '#' immediately inside the directive  
    tag comments out the entire directive
```

```
%]
```

NAME

task_inject.pl – ShakeCast Task Schedule Tool

SYNOPSIS

```
task_inject.pl task [ option ... ]
```

DESCRIPTION

The **task_inject.pl** utility is used to manually queue a task into the ShakeCast database. The queued task needs to be recognized by the ShakeCast Dispatcher or it will return a FAILED status.

The script is usually invoked from the administration interface but also can be executed directly. Depending on the task type, additional parameters are read from the command line.

OPTIONS

--task

Specify the type of task to process. Default task type includes 'comp_gmpe', 'logrotate', 'logstats', 'heartbeat', 'gs_json', 'maintain_event', 'facility_fragility_stat', 'facility_regulatory_level', 'facility_feature_shaking', 'screen_shot', or 'map_tile'.

comp_gmpe

Compute theoretical ground motions for facilities of the specified earthquake. Additional event ID is read from the command line.

logrotate

Rotate the ShakeCast log files. No additional parameters are required.

logstat

Generate log statistics plots. No additional parameters are required.

heartbeat

Trigger a ShakeCast heartbeat message. No additional parameters are required.

gs_json

Refresh the USGS earthquake JSON feed and process new earthquakes. No additional parameters are required.

maintain_event

Trigger to maintain the ShakeCast database. Old ShakeMaps without exposure will be purged from the system. No additional parameters are required.

facility_fragility_stat

Trigger the process to compute probabilistic facility fragility. Additional one event ID and one Version parameters are read from the command line.

facility_regulatory_level

Trigger the process to compute exceedance of regulatory levels. This function is specifically design for nuclear power plants. Additional one event ID and one Version parameters are read from the command line. **facility_feature_shaking**

Trigger the process to compute facility feature shaking for the selected earthquake. Additional one event ID and one Version parameters are read from the command line.

screen_shot

Take a screen shot for the selected earthquake and save the output image. Additional one event ID and one Version parameters are read from the command line.

map_tile

Generate image tile overlay to be displayed on the web interface. Additional tile type parameter is read from the command line.

--event

Specify ID and Version of the event to process.

NAME

tester.pl – ShakeCast Test Event Tool

SYNOPSIS

```
tester.pl [ option ... ]
```

DESCRIPTION

The **tester.pl** utility is used to handle ShakeCast test events and includes conversion, listing, and triggering of test events. The script is usually invoked from the administration interface but also can be executed directly. It reads one process type from the command line.

OPTIONS

--type

Specify the type of action to process. Process type is one of 'event_menu', 'new_test', 'create_test', 'inject_next', or 'inject_first'.

event_menu

Output a list of test events available on the system.

new_test

Output a list of actual events on the system that have not been converted into test events.

create_test

Convert the specified event into a test event that can be triggered locally. Require an additional **-key** option. A new data directory for the event will be created under the "test_data" directory with the name of event ID and "_scte" postfix.

inject_first

Trigger a ShakeCast process for the specified test event as a new event. Require an additional **-key** option.

inject_next

Trigger a ShakeCast process for the specified test event as an updated event. Require an additional **-key** option.

--key

Specify ID of the event to process. All information for the processed ShakeMaps will be removed from the ShakeCast database but not downloaded products in the file system.

Appendix D Relating HAZUS Structure Types to Seismic Performance

D.1 Selecting Model Building Type and Code Era

ShakeCast offers structural damage estimation capability adapted from the HAZUS-MH earthquake module (NIBS and FEMA, 2003). For any site of interest, the user begins by selecting from the available HAZUS model building types, of which there are 36 (table D.1). “Model building type” refers to the materials of construction (wood, steel, reinforced concrete, etc.), the system used to transmit earthquake forces from the ground through the building (referred to as the lateral force-resisting system), and sometimes height category (low-rise, mid-rise, and high-rise, which generally correspond to 1-3, 4-7, and 8+ stories, respectively).

The user must also select for each facility its building code era, of which there are four (high code, moderate code, low code, and pre-code; table D.2 and fig. D.1). Code eras reflect important changes in design forces or detailing requirements that matter to the seismic performance of a building. Sixteen combinations of model building type and code era do not exist (for example, high-code unreinforced masonry bearing wall), so in total there are 128 choices for HAZUS model building type and code era. Note that code era is largely a function of location and year built, so in principal ShakeCast could simplify the user’s job of selecting a code era by asking for era of construction (pre-1941, 1941-1975, or post-1975) instead and then looking up the code era via internal GIS database.

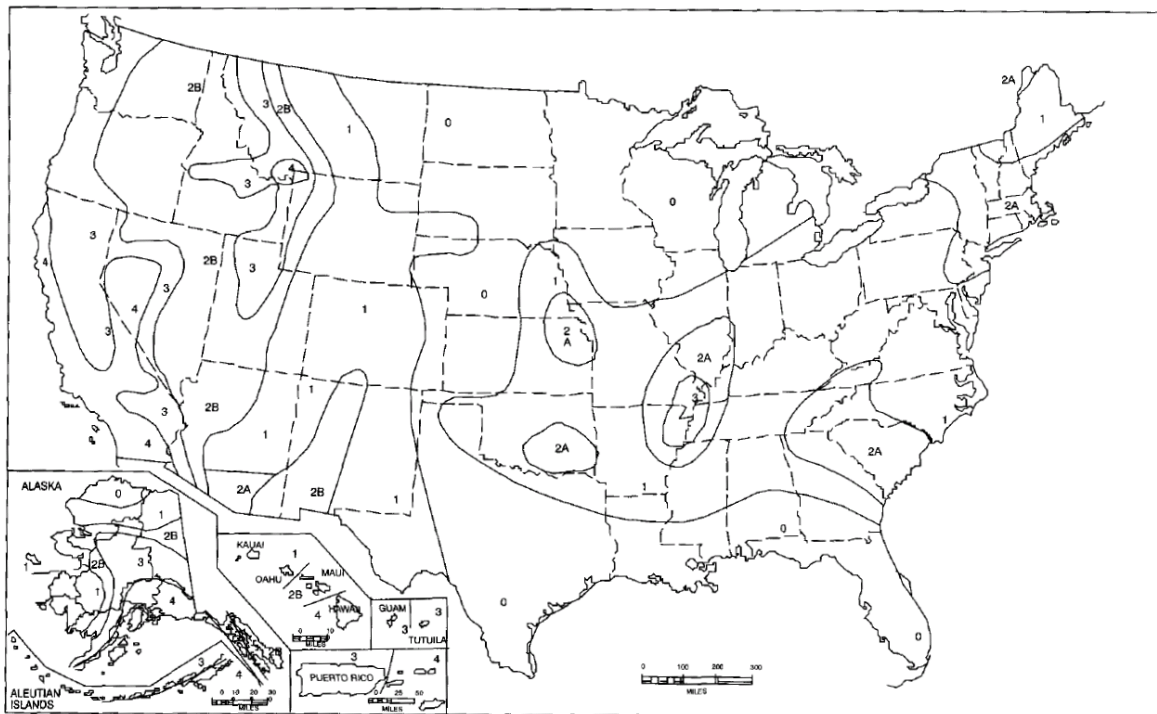


Figure 29. Seismic zone map of the United States (fig. 16-2, ICBO, 1997).

D.2 Describing Potential Damage

The user selects between 3 and 4 alert levels, meaning that any facility affected by an earthquake is noted either green, yellow, or red (3 levels), or green, yellow, orange, or red (4 levels). These colors index the likely structural damage state of the facility in HAZUS terms: green corresponds to HAZUS' undamaged or slight structural damage states, yellow corresponds to moderate structural damage, orange to extensive structural damage, and red to complete structural damage. These terms (slight, moderate, etc.) are described via likely effects of the earthquake on the structural system. For example, for a small wood-frame building (W1, regardless of code era), "green" corresponds to "Undamaged or small plaster or gypsum-board cracks at corners of door and window openings and wall-ceiling intersections; small cracks in masonry chimneys and masonry veneer." These descriptions can be found in the HAZUS-MH technical manual (NIBS and FEMA, 2003) Section 5.3.1.

The code level ("L" for "low") is appended to the structure's Label (e.g., "W1") in order to represent the internal ShakeCast facility type ("W1L") from which a look-up is used to retrieve the corresponding building-specific HAZUS-MH Technical Manual Table 5.16a-d median (alpha) and variability (beta) fragility values.

D.3 Relating Seismic Excitation to Structural Damage

When an earthquake occurs, its shaking intensity at each facility location is estimated in terms of peak horizontal ground acceleration (PGA). Buildings and ground motions are highly variable, even given a model building type and PGA level, so it is uncertain the exact level of PGA that will cause a given facility to experience structural damage of any particular level. The relationship between PGA and damage state is therefore probabilistic, meaning that one can estimate the probability of a given building experiencing a given structural damage state when the building experiences a certain level of PGA. It is more convenient here to estimate the PGA at which there is a given probability of damage exceeding a given structural damage state. In ShakeCast, a facility is indicated as damage level x (that is, green, yellow, orange, or red) when the PGA is such that there is at least a 50% probability of the corresponding HAZUS structural damage state and less than a 50% probability of the next-higher HAZUS structural damage state. These PGA values are taken from the HAZUS-MH Technical Manual Table 5.16a-d.

D.4 Tabular Lookup Data

Two lookup files in CSV format are provided with this manual, one for a three-level damage scheme; the other is for a four-level damage scheme. Each has seven columns or fields, listed in table D.3. The fields correspond to data appearing in the ShakeCast Facility Administration screen (see Section 4.3).

Table 6. HAZUS-MH earthquake model building types (NIBS and FEMA, 2003, Table 3.1)

No.	Label	Description	Height			
			Range		Typical	
			Name	Stories	Stories	Feet
1	W1	Wood, Light Frame ($\leq 5,000$ sq. ft.)		1 - 2	1	14
2	W2			All	2	24
3	S1L	Steel Moment Frame	Low-Rise	1 - 3	2	24
4	S1M		Mid-Rise	4 - 7	5	60
5	S1H		High-Rise	8+	13	156
6	S2L	Steel Braced Frame	Low-Rise	1 - 3	2	24
7	S2M		Mid-Rise	4 - 7	5	60
8	S2H		High-Rise	8+	13	156
9	S3	Steel Light Frame		All	1	15
10	S4L	Steel Frame with Cast-in-Place Concrete Shear Walls	Low-Rise	1 - 3	2	24
11	S4M		Mid-Rise	4 - 7	5	60
12	S4H		High-Rise	8+	13	156
13	S5L	Steel Frame with Unreinforced Masonry Infill Walls	Low-Rise	1 - 3	2	24
14	S5M		Mid-Rise	4 - 7	5	60 156
15	S5H		High-Rise	8+	13	
16	C1L	Concrete Moment Frame	Low-Rise	1 - 3	2	20
17	C1M		Mid-Rise	4 - 7	5	50
18	C1H		High-Rise	8+	12	120
19	C2L	Concrete Shear Walls	Low-Rise	1 - 3	2	20
20	C2M		Mid-Rise	4 - 7	5	50
21	C2H		High-Rise	8+	12	120
22	C3L	Concrete Frame with Unreinforced Masonry Infill Walls	Low-Rise	1 - 3	2	20
23	C3M		Mid-Rise	4 - 7	5	50
24	C3H		High-Rise	8+	12	120
25	PC1	Precast Concrete Tilt-Up Walls		All	1	15
26	PC2L	Precast Concrete Frames with Concrete Shear Walls	Low-Rise	1 - 3	2	20
27	PC2M		Mid-Rise	4 - 7	5	50
28	PC2H		High-Rise	8+	12	120
29	RM1L	Reinforced Masonry Bearing Walls with Wood or Metal Deck Diaphragms	Low-Rise	1-3	2	20
30	RM2M		Mid-Rise	4+	5	50
31	RM2L	Reinforced Masonry Bearing Walls with Precast Concrete Diaphragms	Low-Rise	1 - 3	2	20
32	RM2M		Mid-Rise	4 - 7	5	50
33	RM2H		High-Rise	8+	12	120
34	URML	Unreinforced Masonry Bearing Walls	Low-Rise	1 - 2	1	15
35	URMM		Mid-Rise	3+	3	35
36	MH	Mobile Homes		All	1	10

Table 7. HAZUS-MH guidelines for selection of damage functions for typical buildings based on UBC seismic zone and building age (NIBS and FEMA, 2003, Table 5.20).

UBC Seismic Zone (NEHRP Map Area)	Post-1975	1941 - 1975	Pre-1941
Zone 4 (Map Area 7)	High-Code	Moderate-Code	Pre-Code (W1 = Moderate-Code)
Zone 3 (Map Area 6)	Moderate-Code	Moderate-Code	Pre-Code (W1 = Moderate-Code)
Zone 2B (Map Area 5)	Moderate-Code	Low-Code	Pre-Code (W1 = Low-Code)
Zone 2A (Map Area 4)	Low-Code	Low-Code	Pre-Code (W1 = Low-Code)
Zone 1 (Map Area 2/3)	Low-Code	Pre-Code (W1 = Low-Code)	Pre-Code (W1 = Low-Code)
Zone 0 (Map Area 1)	Pre-Code (W1 = Low-Code)	Pre-Code (W1 = Low-Code)	Pre-Code (W1 = Low-Code)

Table 8. Layout of damage lookup tables.

Field name	Type	Description
ID	Integer	A unique index
Facility Type	String	HAZUS model building type and seismic design level
Color	String	Green, Yellow, Orange, or Red
Damage Level	String	Equivalent HAZUS structural damage level(s)
Low Limit	Integer	Intensity with 50% probability of this damage level occurring
High Limit	Integer	Intensity with 50% probability of next damage level occurring
Metric	String	Intensity metric

Appendix E HAZUS Model Building Fragility Table

Table 9. High-code fragilities for HAZUS model building types.

FACILITY TYPE	DESCRIPTION	METRIC	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA
W1H	Wood, Light Frame	PGA	26	0.64	55	0.64	128	0.64	201	0.64
W2H	Wood Commercial	PGA	26	0.64	56	0.64	115	0.64	208	0.64
S1LH	Steel Moment Frame, Low Rise	PGA	19	0.64	31	0.64	64	0.64	149	0.64
S1MH	Steel Moment Frame, Mid Rise	PGA	14	0.64	26	0.64	62	0.64	143	0.64
S1HH	Steel Moment Frame, High Rise	PGA	10	0.64	21	0.64	52	0.64	131	0.64
S2LH	Steel Brace Frame, Low Rise	PGA	24	0.64	41	0.64	76	0.64	146	0.64
S2MH	Steel Brace Frame, Mid Rise	PGA	14	0.64	27	0.64	73	0.64	162	0.64
S2HH	Steel Brace Frame, High Rise	PGA	11	0.64	22	0.64	65	0.64	160	0.64
S3H	Steel Light Frame	PGA	15	0.64	26	0.64	54	0.64	100	0.64
S4LH	Steel Frame Shear Wall, Low Rise	PGA	24	0.64	39	0.64	71	0.64	133	0.64
S4MH	Steel Frame Shear Wall, Mid Rise	PGA	16	0.64	28	0.64	73	0.64	156	0.64
S4HH	Steel Frame Shear Wall, High Rise	PGA	13	0.64	25	0.64	69	0.64	163	0.64
C1LH	Concrete Moment Frame, Low Rise	PGA	21	0.64	35	0.64	70	0.64	137	0.64
C1MH	Concrete Moment Frame, Mid Rise	PGA	15	0.64	27	0.64	73	0.64	161	0.64
C1HH	Concrete Moment Frame, High Rise	PGA	11	0.64	22	0.64	62	0.64	135	0.64
C2LH	Concrete Shear Wall, Low Rise	PGA	24	0.64	45	0.64	90	0.64	155	0.64
C2MH	Concrete Shear Wall, Mid Rise	PGA	17	0.64	36	0.64	87	0.64	195	0.64
C2HH	Concrete Shear Wall, High Rise	PGA	12	0.64	29	0.64	82	0.64	187	0.64
PC1H	Precast Concrete Tiltup Wall	PGA	20	0.64	35	0.64	72	0.64	125	0.64
PC2LH	Precast Concrete Frames, Low Rise	PGA	24	0.64	36	0.64	69	0.64	123	0.64
PC2MH	Precast Concrete Frames, Mid Rise	PGA	17	0.64	29	0.64	67	0.64	151	0.64
PC2HH	Precast Concrete Frames, High Rise	PGA	12	0.64	23	0.64	63	0.64	149	0.64
RM1LH	Reinforced Masonry with Wood, Low Rise	PGA	30	0.64	46	0.64	93	0.64	157	0.64
RM1MH	Reinforced Masonry with Wood, Mid Rise	PGA	20	0.64	37	0.64	81	0.64	190	0.64
RM2LH	Reinforced Masonry with Precast Concrete, Low Rise	PGA	26	0.64	42	0.64	87	0.64	149	0.64

FACILITY TYPE	DESCRIPTION	METRIC	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA
RM2MH	Reinforced Masonry with Precast Concrete, Mid Rise	PGA	17	0.64	33	0.64	75	0.64	183	0.64
RM2HH	Reinforced Masonry with Precast Concrete, High Rise	PGA	12	0.64	24	0.64	67	0.64	178	0.64
MHH	Mobile Homes	PGA	11	0.64	18	0.64	31	0.64	60	0.64

Table 10. Moderate-code fragilities for HAZUS model building types.

FACILITY TYPE	DESCRIPTION	METRIC	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA
W1M	Wood, Light Frame	PGA	24	0.64	43	0.64	91	0.64	134	0.64
W2M	Wood Commercial	PGA	20	0.64	35	0.64	64	0.64	113	0.64
S1LM	Steel Moment Frame, Low Rise	PGA	15	0.64	22	0.64	42	0.64	80	0.64
S1MM	Steel Moment Frame, Mid Rise	PGA	13	0.64	21	0.64	44	0.64	82	0.64
S1HM	Steel Moment Frame, High Rise	PGA	10	0.64	18	0.64	39	0.64	78	0.64
S2LM	Steel Brace Frame, Low Rise	PGA	20	0.64	26	0.64	46	0.64	84	0.64
S2MM	Steel Brace Frame, Mid Rise	PGA	14	0.64	22	0.64	53	0.64	97	0.64
S2HM	Steel Brace Frame, High Rise	PGA	11	0.64	19	0.64	49	0.64	102	0.64
S3M	Steel Light Frame	PGA	13	0.64	19	0.64	33	0.64	60	0.64
S4LM	Steel Frame Shear Wall, Low Rise	PGA	19	0.64	26	0.64	41	0.64	78	0.64
S4MM	Steel Frame Shear Wall, Mid Rise	PGA	14	0.64	22	0.64	51	0.64	92	0.64
S4HM	Steel Frame Shear Wall, High Rise	PGA	12	0.64	21	0.64	51	0.64	97	0.64
C1LM	Concrete Moment Frame, Low Rise	PGA	16	0.64	23	0.64	41	0.64	77	0.64
C1MM	Concrete Moment Frame, Mid Rise	PGA	13	0.64	21	0.64	49	0.64	89	0.64
C1HM	Concrete Moment Frame, High Rise	PGA	11	0.64	18	0.64	41	0.64	74	0.64
C2LM	Concrete Shear Wall, Low Rise	PGA	18	0.64	30	0.64	49	0.64	87	0.64
C2MM	Concrete Shear Wall, Mid Rise	PGA	15	0.64	26	0.64	55	0.64	102	0.64
C2HM	Concrete Shear Wall, High Rise	PGA	12	0.64	23	0.64	57	0.64	107	0.64
PC1M	Precast Concrete Tiltup Wall	PGA	18	0.64	24	0.64	44	0.64	71	0.64
PC2LM	Precast Concrete Frames, Low Rise	PGA	18	0.64	25	0.64	40	0.64	74	0.64

FACILITY TYPE	DESCRIPTION	METRIC	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA
PC2MM	Precast Concrete Frames, Mid Rise	PGA	15	0.64	21	0.64	45	0.64	86	0.64
PC2HM	Precast Concrete Frames, High Rise	PGA	12	0.64	19	0.64	46	0.64	90	0.64
RM1LM	Reinforced Masonry with Wood, Low Rise	PGA	22	0.64	30	0.64	50	0.64	85	0.64
RM1MM	Reinforced Masonry with Wood, Mid Rise	PGA	18	0.64	26	0.64	51	0.64	103	0.64
RM2LM	Reinforced Masonry with Precast Concrete, Low Rise	PGA	20	0.64	28	0.64	47	0.64	81	0.64
RM2MM	Reinforced Masonry with Precast Concrete, Mid Rise	PGA	16	0.64	23	0.64	48	0.64	99	0.64
RM2HM	Reinforced Masonry with Precast Concrete, High Rise	PGA	12	0.64	20	0.64	48	0.64	101	0.64
MHM	Mobile Homes	PGA	11	0.64	18	0.64	31	0.64	60	0.64

Table 11. Low-code fragilities for HAZUS model building types.

FACILITY TYPE	DESCRIPTION	METRIC	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA
W1L	Wood, Light Frame	PGA	20	0.64	34	0.64	61	0.64	95	0.64
W2L	Wood Commercial	PGA	14	0.64	23	0.64	48	0.64	75	0.64
S1LL	Steel Moment Frame, Low Rise	PGA	12	0.64	17	0.64	30	0.64	48	0.64
S1ML	Steel Moment Frame, Mid Rise	PGA	12	0.64	18	0.64	29	0.64	49	0.64
S1HL	Steel Moment Frame, High Rise	PGA	10	0.64	15	0.64	28	0.64	48	0.64
S2LL	Steel Brace Frame, Low Rise	PGA	13	0.64	17	0.64	30	0.64	50	0.64
S2ML	Steel Brace Frame, Mid Rise	PGA	12	0.64	18	0.64	35	0.64	58	0.64
S2HL	Steel Brace Frame, High Rise	PGA	11	0.64	17	0.64	36	0.64	63	0.64
S3L	Steel Light Frame	PGA	10	0.64	13	0.64	20	0.64	38	0.64
S4LL	Steel Frame Shear Wall, Low Rise	PGA	13	0.64	16	0.64	26	0.64	46	0.64
S4ML	Steel Frame Shear Wall, Mid Rise	PGA	12	0.64	17	0.64	31	0.64	54	0.64
S4HL	Steel Frame Shear Wall, High Rise	PGA	12	0.64	17	0.64	33	0.64	59	0.64
S5LL	Steel Frame Unreinforced Wall, Low Rise	PGA	13	0.64	17	0.64	28	0.64	45	0.64
S5ML	Steel Frame Unreinforced Wall, Mid Rise	PGA	11	0.64	18	0.64	34	0.64	53	0.64
S5HL	Steel Frame Unreinforced Wall, High Rise	PGA	10	0.64	18	0.64	35	0.64	58	0.64
C1LL	Concrete Moment Frame, Low Rise	PGA	12	0.64	15	0.64	27	0.64	45	0.64

FACILITY TYPE	DESCRIPTION	METRIC	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA
C1ML	Concrete Moment Frame, Mid Rise	PGA	12	0.64	17	0.64	32	0.64	54	0.64
C1HL	Concrete Moment Frame, High Rise	PGA	10	0.64	15	0.64	27	0.64	44	0.64
C2LL	Concrete Shear Wall, Low Rise	PGA	14	0.64	19	0.64	30	0.64	52	0.64
C2ML	Concrete Shear Wall, Mid Rise	PGA	12	0.64	19	0.64	38	0.64	63	0.64
C2HL	Concrete Shear Wall, High Rise	PGA	11	0.64	19	0.64	38	0.64	65	0.64
C3LL	Concrete Unreinforced Wall, Low Rise	PGA	12	0.64	17	0.64	26	0.64	44	0.64
C3ML	Concrete Unreinforced Wall, Mid Rise	PGA	11	0.64	17	0.64	32	0.64	51	0.64
C3HL	Concrete Unreinforced Wall, High Rise	PGA	9	0.64	16	0.64	33	0.64	53	0.64
PC1L	Precast Concrete Tiltup Wall	PGA	13	0.64	17	0.64	25	0.64	45	0.64
PC2LL	Precast Concrete Frames, Low Rise	PGA	13	0.64	15	0.64	24	0.64	44	0.64
PC2ML	Precast Concrete Frames, Mid Rise	PGA	11	0.64	16	0.64	31	0.64	52	0.64
PC2HL	Precast Concrete Frames, High Rise	PGA	11	0.64	16	0.64	31	0.64	55	0.64
RM1LL	Reinforced Masonry with Wood, Low Rise	PGA	16	0.64	20	0.64	29	0.64	54	0.64
RM1ML	Reinforced Masonry with Wood, Mid Rise	PGA	14	0.64	19	0.64	35	0.64	63	0.64
RM2LL	Reinforced Masonry with Precast Concrete, Low Rise	PGA	14	0.64	18	0.64	28	0.64	51	0.64
RM2ML	Reinforced Masonry with Precast Concrete, Mid Rise	PGA	12	0.64	17	0.64	34	0.64	60	0.64
RM2HL	Reinforced Masonry with Precast Concrete, High Rise	PGA	11	0.64	17	0.64	35	0.64	62	0.64
URMLL	Unreinforced Masonry, Low Rise	PGA	14	0.64	20	0.64	32	0.64	46	0.64
URMML	Unreinforced Masonry, Mid Rise	PGA	10	0.64	16	0.64	27	0.64	46	0.64
MHL	Mobile Homes	PGA	11	0.64	18	0.64	31	0.64	60	0.64

Table 12. Pre-code fragilities for HAZUS model building types.

FACILITY TYPE	DESCRIPTION	METRIC	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA
W1P	Wood, Light Frame	PGA	18	0.64	29	0.64	51	0.64	77	0.64
W2P	Wood Commercial	PGA	12	0.64	19	0.64	37	0.64	60	0.64
S1LP	Steel Moment Frame, Low Rise	PGA	9	0.64	13	0.64	22	0.64	38	0.64

FACILITY TYPE	DESCRIPTION	METRIC	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA
S1MP	Steel Moment Frame, Mid Rise	PGA	9	0.64	14	0.64	23	0.64	39	0.64
S1HP	Steel Moment Frame, High Rise	PGA	8	0.64	12	0.64	22	0.64	38	0.64
S2LP	Steel Brace Frame, Low Rise	PGA	11	0.64	14	0.64	23	0.64	39	0.64
S2MP	Steel Brace Frame, Mid Rise	PGA	10	0.64	14	0.64	28	0.64	47	0.64
S2HP	Steel Brace Frame, High Rise	PGA	9	0.64	13	0.64	29	0.64	50	0.64
S3P	Steel Light Frame	PGA	8	0.64	10	0.64	16	0.64	30	0.64
S4LP	Steel Frame Shear Wall, Low Rise	PGA	10	0.64	13	0.64	20	0.64	36	0.64
S4MP	Steel Frame Shear Wall, Mid Rise	PGA	9	0.64	13	0.64	25	0.64	43	0.64
S4HP	Steel Frame Shear Wall, High Rise	PGA	9	0.64	14	0.64	27	0.64	47	0.64
S5LP	Steel Frame Unreinforced Wall, Low Rise	PGA	11	0.64	14	0.64	22	0.64	37	0.64
S5MP	Steel Frame Unreinforced Wall, Mid Rise	PGA	9	0.64	14	0.64	28	0.64	43	0.64
S5HP	Steel Frame Unreinforced Wall, High Rise	PGA	8	0.64	14	0.64	29	0.64	46	0.64
C1LP	Concrete Moment Frame, Low Rise	PGA	10	0.64	12	0.64	21	0.64	36	0.64
C1MP	Concrete Moment Frame, Mid Rise	PGA	9	0.64	13	0.64	26	0.64	43	0.64
C1HP	Concrete Moment Frame, High Rise	PGA	8	0.64	12	0.64	21	0.64	35	0.64
C2LP	Concrete Shear Wall, Low Rise	PGA	11	0.64	15	0.64	24	0.64	42	0.64
C2MP	Concrete Shear Wall, Mid Rise	PGA	10	0.64	15	0.64	30	0.64	50	0.64
C2HP	Concrete Shear Wall, High Rise	PGA	9	0.64	15	0.64	31	0.64	52	0.64
C3LP	Concrete Unreinforced Wall, Low Rise	PGA	10	0.64	14	0.64	21	0.64	35	0.64
C3MP	Concrete Unreinforced Wall, Mid Rise	PGA	9	0.64	14	0.64	25	0.64	41	0.64
C3HP	Concrete Unreinforced Wall, High Rise	PGA	8	0.64	13	0.64	27	0.64	43	0.64
PC1P	Precast Concrete Tiltup Wall	PGA	11	0.64	14	0.64	21	0.64	35	0.64
PC2LP	Precast Concrete Frames, Low Rise	PGA	10	0.64	13	0.64	19	0.64	35	0.64
PC2MP	Precast Concrete Frames, Mid Rise	PGA	9	0.64	13	0.64	24	0.64	42	0.64
PC2HP	Precast Concrete Frames, High Rise	PGA	9	0.64	13	0.64	25	0.64	43	0.64
RM1LP	Reinforced Masonry with Wood, Low Rise	PGA	13	0.64	16	0.64	24	0.64	43	0.64
RM1MP	Reinforced Masonry with Wood, Mid Rise	PGA	11	0.64	15	0.64	28	0.64	50	0.64
RM2LP	Reinforced Masonry with Precast Concrete, Low Rise	PGA	12	0.64	15	0.64	22	0.64	41	0.64

FACILITY TYPE	DESCRIPTION	METRIC	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA	ALPHA	BETA
RM2MP	Reinforced Masonry with Precast Concrete, Mid Rise	PGA	10	0.64	14	0.64	26	0.64	47	0.64
RM2HP	Reinforced Masonry with Precast Concrete, High Rise	PGA	9	0.64	13	0.64	27	0.64	50	0.64
URMLP	Unreinforced Masonry, Low Rise	PGA	13	0.64	17	0.64	26	0.64	37	0.64
URMMP	Unreinforced Masonry, Mid Rise	PGA	9	0.64	13	0.64	21	0.64	38	0.64
MHP	Mobile Homes	PGA	8	0.64	11	0.64	18	0.64	34	0.64

Appendix F ShakeCast Notification Template Keywords

Table 13. EVENT Notification Keywords

Constant	Description
EVENT_ID	Earthquake identifier e.g., nn00423851
EVENT_VERSION	Integer indicating event version
EVENT_STATUS	ShakeMap status NORMAL: RELEASED REVIEWED CANCELLED:
EVENT_NAME	String name describing event; defined by local network
MAGNITUDE	Event magnitude
EVENT_LOCATION_DESCRIPTION	String name describing event location with geographic reference; defined by local network e.g., "32km WNW of Alamo, Nevada"
EVENT_TIMESTAMP	Event timestamp e.g., yyyy-mm-ddThh:mm:ssZ
LAT	Event latitude (decimal degrees, north)
LON	Event longitude (decimal degrees, east)
EXTERNAL_EVENT_ID	Same as EVENT_ID in V3
NOTIFICATION_ID	Sequence ID in ShakeCast notification table
TRIES	Number of notification attempts.
DELIVERY_STATUS	Result of notification attempt.
SHAKECAST_USER	ShakeCast User ID
DELIVERY_ADDRESS	Email address for delivery
DELIVERY_METHOD	Product delivery type for the given notification EMAIL_HTML: email with html formatting EMAIL_TEXT: plain text email PAGER: simple text message for SMS delivery
EVENT_TYPE	Types of events that will trigger notifications to be sent: ALL: includes all event types (inclusive) ACTUAL: real earthquake SCENARIO: a scenario or converted actual event TEST: system test HEARTBEAT: a heartbeat system test message
NOTIFICATION_TYPE	Types of events that will trigger a notification to be sent

Constant	Description
	<p>Valid notification types:</p> <p>NEW_EVENT: an earthquake exceeding a user-set threshold value. Requires EVENT_TYPE and DELIVERY_METHOD tags.</p> <p>DAMAGE: Triggered when the ground shaking parameter at a facility (or facilities) is between the high and low values of the user-set facility parameters. Requires EVENT_TYPE, DELIVERY_METHOD, and DAMAGE_LEVEL tags.</p> <p>SHAKING: Triggered when the ground shaking parameter at the facility location of the facility exceeds the preset value. Requires EVENT_TYPE, DELIVERY_METHOD, METRIC, and LIMIT_VALUE tags.</p> <p>CAN_EVENT: cancelled event. Requires EVENT_TYPE and DELIVERY_METHOD tags.</p> <p>UPD_EVENT: updated event. Requires EVENT_TYPE and DELIVERY_METHOD tags.</p> <p>NEW_PROD: triggered when a specific ShakeMap product becomes available. Require EVENT_TYPE, DELIVERY_METHOD, and PRODUCT tags.</p>
MESSAGE_FORMAT	Name of notification template (default)
LIMIT_VALUE	Minimum magnitude for a notification to be sent
PRODUCT_TYPE	Type of product to be delivered. If omitted, product is plain text. PDF: PDF from templates
FILENAME	External files to be attached to message
AGGREGATION_GROUP	Notification to be sent based on GROUP type defined by membership in a GROUP. ShakeCast has a predefined CITY group of global cities.
MAX_TRIES	Maximum number of notification attempts.
FACILITY_ID	Unique facility identifier. Text(32)
FACILITY_TYPE	Type of facility. Current defined types are: BRIDGE, CAMPUS, CITY, COUNTY, DAM, DISTRICT, ENGINEERED, INDUSTRIAL, MULTIFAM, ROAD, SINGLEFAM, STRUCTURE, TANK, TUNNEL, UNKNOWN, and HAZUS building types.
EXTERNAL_FACILITY_ID	Organization's unique facility identifier. Text (32) This field must be unique for a facility type but the same external_facility_id may be used for different types of facilities.
FACILITY_NAME	Facility name. Text(128). The value of this field is displayed to the user.
SHORT_NAME	Shortened version of facility name. Text(128).

Constant	Description
	ShakeCast uses the value in this field when a shorter version of the name is needed due to output space limitations.
DESCRIPTION	Facility description. Text(255).
FACILITY_LAT	Facility latitude (decimal degrees, north)
FACILITY_LON	Facility longitude (decimal degrees, east)
GEOM_TYPE	The value of this field is used by ShakeCast to handle the geometry coordinates from the geom field. Text(32) Currently defined types are: POINT, POLYLINE, POLYGON, RECTANGLE, and CIRCLE.
GEOM	Geometry of a facility. The value of this field is used specify the coordinates of the facility. Text(32) Format of this field is in (longitude latitude) pairs separated by a white space. The size limit of data is ~16MB.
AGGREGATE	Flag to indicate whether notifications should be combined into a single message Integer value e.g., 1
SERVER_ID	Identifier of this ShakeCast server
DNS_ADDRESS	Domain name of this ShakeCast server

Table 14. PRODUCT Notification Keywords

Constant	Description
PRODUCT_ID	Product sequence ID in ShakeCast database
PRODUCT_TYPE	Type of product to be delivered. If omitted, product is plain text. PDF: PDF from templates
NAME	Short product type description
DESCRIPTION	Long product type description
FILENAME	Filename for the product on local system
GENERATION_TIMESTAMP	Timestamp showing when the product was created on remote or local server
PRODUCT_STATUS	Status of product as RELEASED, REVIEWED, or CANCELLED
EVENT_ID	Earthquake identifier e.g., nn00423851
EVENT_VERSION	Integer indicating event version
EVENT_NAME	String name describing event; defined by local network
MAGNITUDE	Event magnitude
EVENT_LOCATION_DESCRIPTION	String name describing event location with geographic reference;

Constant	Description
	defined by local network e.g., "32km WNW of Alamo, Nevada"
EVENT_TIMESTAMP	Event timestamp e.g., yyyy-mm-ddThh:mm:ssZ
LAT	Event latitude (decimal degrees, north)
LON	Event longitude (decimal degrees, east)
SHAKEMAP_ID	Same as the event id
SHAKEMAP_VERSION	Integer indicating map revision
NOTIFICATION_ID	Sequence ID in ShakeCast notification table
TRIES	Number of notification attempts.
DELIVERY_STATUS	Result of notification attempt.
SHAKECAST_USER	ShakeCast User ID
DELIVERY_ADDRESS	Email address for delivery
DELIVERY_METHOD	Product delivery type for the given notification EMAIL_HTML: email with html formatting EMAIL_TEXT: plain text email PAGER: simple text message for SMS delivery
NOTIFICATION_TYPE	Types of events that will trigger a notification to be sent Valid notification types: NEW_EVENT: an earthquake exceeding a user-set threshold value. Requires EVENT_TYPE and DELIVERY_METHOD tags. DAMAGE: Triggered when the ground shaking parameter at a facility (or facilities) is between the high and low values of the user-set facility parameters. Requires EVENT_TYPE, DELIVERY_METHOD, and DAMAGE_LEVEL tags. SHAKING: Triggered when the ground shaking parameter at the facility location of the facility exceeds the preset value. Requires EVENT_TYPE, DELIVERY_METHOD, METRIC, and LIMIT_VALUE tags. CAN_EVENT: cancelled event. Requires EVENT_TYPE and DELIVERY_METHOD tags. UPD_EVENT: updated event. Requires EVENT_TYPE and DELIVERY_METHOD tags. NEW_PROD: triggered when a specific ShakeMap product becomes available. Require EVENT_TYPE, DELIVERY_METHOD, and PRODUCT tags.
MESSAGE_FORMAT	Name of notification template (default)
LIMIT_VALUE	Minimum magnitude for a notification to be sent

Constant	Description
AGGREGATION_GROUP	Notification to be sent based on GROUP type defined by membership in a GROUP. ShakeCast has a predefined CITY group of global cities.
MAX_TRIES	Maximum number of notification attempts.
FACILITY_ID	Unique facility identifier. Text(32)
FACILITY_TYPE	Type of facility. Current defined types are: BRIDGE, CAMPUS, CITY, COUNTY, DAM, DISTRICT, ENGINEERED, INDUSTRIAL, MULTIFAM, ROAD, SINGLEFAM, STRUCTURE, TANK, TUNNEL, UNKNOWN, and HAZUS building types.
EXTERNAL_FACILITY_ID	Organization's unique facility identifier. Text (32) This field must be unique for a facility type but the same external_facility_id may be used for different types of facilities.
FACILITY_NAME	Facility name. Text(128). The value of this field is displayed to the user.
SHORT_NAME	Shortened version of facility name. Text(128). ShakeCast uses the value in this field when a shorter version of the name is needed due to output space limitations.
DESCRIPTION	Facility description. Text(255).
FACILITY_LAT	Facility latitude (decimal degrees, north)
FACILITY_LON	Facility longitude (decimal degrees, east)
GEOM_TYPE	The value of this field is used by ShakeCast to handle the geometry coordinates from the geom field. Text(32) Currently defined types are: POINT, POLYLINE, POLYGON, RECTANGLE, and CIRCLE.
GEOM	Geometry of a facility. The value of this field is used specify the coordinates of the facility. Text(32) Format of this field is in (longitude latitude) pairs separated by a white space. The size limit of data is ~16MB.
AGGREGATE	Flag to indicate whether notifications should be combined into a single message Integer value e.g., 1
SERVER_ID	Identifier of this ShakeCast server
DNS_ADDRESS	Domain name of this ShakeCast server

Table 15. SHAKING Notification Keywords

Constant	Description
SHAKEMAP_ID	Same as the event id
SHAKEMAP_VERSION	Integer indicating map revision
SHAKEMAP_REGION	ShakeMap Network Code
GENERATION_TIMESTAMP	ShakeCast processing timestamp e.g., yyyy-mm-ddThh:mm:ssZ
EVENT_ID	Earthquake identifier e.g., nn00423851
EVENT_VERSION	Integer indicating event version Integer indicating event version
EVENT_NAME	String name describing event; defined by local network
MAGNITUDE	Event magnitude
EVENT_LOCATION_DESCRIPTION	String name describing event location with geographic reference; defined by local network e.g., "32km WNW of Alamo, Nevada"
EVENT_TIMESTAMP	Event timestamp e.g., yyyy-mm-ddThh:mm:ssZ
LAT	Event latitude (decimal degrees, north)
LON	Event longitude (decimal degrees, east)
NOTIFICATION_ID	Sequence ID in ShakeCast notification table
TRIES	Number of notification attempts.
DELIVERY_STATUS	Result of notification attempt.
DELIVERY_ADDRESS	Email address for delivery
METRIC	ShakeMap metric for the shaking value
GRID_VALUE	ShakeMap shaking value
DELIVERY_METHOD	Product delivery type for the given notification EMAIL_HTML: email with html formatting EMAIL_TEXT: plain text email PAGER: simple text message for SMS delivery
NOTIFICATION_TYPE	Types of events that will trigger a notification to be sent Valid notification types: NEW_EVENT: an earthquake exceeding a user-set threshold value. Requires EVENT_TYPE and DELIVERY_METHOD tags. DAMAGE: Triggered when the ground shaking parameter at a facility (or facilities) is between the high and low values of the user-set facility parameters. Requires EVENT_TYPE, DELIVERY_METHOD, and DAMAGE_LEVEL tags. SHAKING: Triggered when the ground shaking parameter at the facility location of the facility exceeds the preset value. Requires

Constant	Description
	<p>EVENT_TYPE, DELIVERY_METHOD, METRIC, and LIMIT_VALUE tags.</p> <p>CAN_EVENT: cancelled event. Requires EVENT_TYPE and DELIVERY_METHOD tags.</p> <p>UPD_EVENT: updated event. Requires EVENT_TYPE and DELIVERY_METHOD tags.</p> <p>NEW_PROD: triggered when a specific ShakeMap product becomes available. Require EVENT_TYPE, DELIVERY_METHOD, and PRODUCT tags.</p>
MESSAGE_FORMAT	Name of notification template (default)
LIMIT_VALUE	Minimum magnitude for a notification to be sent
PRODUCT_TYPE	<p>Type of product to be delivered by the specified DELIVERY_METHOD.</p> <p>Products include</p> <p>PDF:</p> <p>GRID_XML</p> <p>PGA_JPG</p> <p>INTEN_JPG</p>
FILENAME	External file to be attached to message
AGGREGATION_GROUP	Notification to be sent based on GROUP type defined by membership in a GROUP. ShakeCast has a predefined CITY group of global cities.
MAX_TRIES	Maximum number of notification attempts.
FACILITY_ID	Unique facility identifier. Text(32)
FACILITY_TYPE	Type of facility. Current defined types are: BRIDGE, CAMPUS, CITY, COUNTY, DAM, DISTRICT, ENGINEERED, INDUSTRIAL, MULTIFAM, ROAD, SINGLEFAM, STRUCTURE, TANK, TUNNEL, UNKNOWN, and HAZUS building types.
EXTERNAL_FACILITY_ID	<p>Organization's unique facility identifier. Text (32)</p> <p>This field must be unique for a facility type but the same external_facility_id may be used for different types of facilities.</p>
FACILITY_NAME	<p>Facility name. Text(128).</p> <p>The value of this field is displayed to the user.</p>
SHORT_NAME	<p>Shortened version of facility name. Text(128).</p> <p>ShakeCast uses the value in this field when a shorter version of the name is needed due to output space limitations.</p>
DESCRIPTION	Facility description. Text(255).
FACILITY_LAT	Facility latitude (decimal degrees, north)
FACILITY_LON	Facility longitude (decimal degrees, east)

Constant	Description
GEOM_TYPE	The value of this field is used by ShakeCast to handle the geometry coordinates from the geom field. Text(32) Currently defined types are: POINT, POLYLINE, POLYGON, RECTANGLE, and CIRCLE.
GEOM	Geometry of a facility. The value of this field is used specify the coordinates of the facility. Text(32) Format of this field is in (longitude latitude) pairs separated by a white space. The size limit of data is ~16MB.
BOUND_SOUTH	ShakeMap boundary to south
BOUND_NORTH	ShakeMap boundary to north
BOUND_WEST	ShakeMap boundary to west
BOUND_EAST	ShakeMap boundary to east
AGGREGATE	Flag to indicate whether notifications should be combined into a single message Integer value e.g., 1
SERVER_ID	Identifier of this ShakeCast server
DNS_ADDRESS	Domain name of this ShakeCast server

Table 16. DAMAGE Notification Keywords

Constant	Description
SHAKEMAP_ID	Same as the event id
SHAKEMAP_VERSION	Integer indicating map revision
SHAKEMAP_REGION	ShakeMap network code
GENERATION_TIMESTAMP	ShakeCast processing timestamp e.g., yyyy-mm-ddThh:mm:ssZ
EVENT_ID	Earthquake identifier e.g., nn00423851
EVENT_VERSION	Integer indicating event version
EVENT_NAME	String name describing event; defined by local network
MAGNITUDE	Event magnitude
EVENT_LOCATION_DESCRIPTION	String name describing event location with geographic reference; defined by local network e.g., "32km WNW of Alamo, Nevada"
EVENT_TIMESTAMP	Event timestamp e.g., yyyy-mm-ddThh:mm:ssZ
LAT	Event latitude (decimal degrees, north)
LON	Event longitude (decimal degrees, north)

Constant	Description
NOTIFICATION_ID	Sequence ID in ShakeCast notification table
TRIES	Number of notification attempts.
DELIVERY_STATUS	Result of notification attempt.
SHAKECAST_USER	ShakeCast User ID
DELIVERY_ADDRESS	Email address for delivery
METRIC	ShakeMap metric used for damage assessment
GRID_VALUE	ShakeMap value used for damage assessment
DAMAGE_LEVEL	String parameter for notification to be sent within the damage threshold e.g., GREEN, ORANGE, YELLOW, RED
DAMAGE_LEVEL_NAME	Damage level description
IS_MAX_SEVERITY	Flag showing whether this is the most severe damage state
SEVERITY_RANK	Rank of damage state
LOW_LIMIT	Minimum shaking value of the damage state
HIGH_LIMIT	Maximum shaking value of the damage state
DELIVERY_METHOD	Product delivery type for the given notification EMAIL_HTML: email with html formatting EMAIL_TEXT: plain text email PAGER: simple text message for SMS delivery
NOTIFICATION_TYPE	Types of events that will trigger a notification to be sent Valid notification types: NEW_EVENT: an earthquake exceeding a user-set threshold value. Requires EVENT_TYPE and DELIVERY_METHOD tags. DAMAGE: Triggered when the ground shaking parameter at a facility (or facilities) is between the high and low values of the user-set facility parameters. Requires EVENT_TYPE, DELIVERY_METHOD, and DAMAGE_LEVEL tags. SHAKING: Triggered when the ground shaking parameter at the facility location of the facility exceeds the preset value. Requires EVENT_TYPE, DELIVERY_METHOD, METRIC, and LIMIT_VALUE tags. CAN_EVENT: cancelled event. Requires EVENT_TYPE and DELIVERY_METHOD tags. UPD_EVENT: updated event. Requires EVENT_TYPE and DELIVERY_METHOD tags. NEW_PROD: triggered when a specific ShakeMap product becomes available. Require EVENT_TYPE, DELIVERY_METHOD, and PRODUCT tags.

Constant	Description
MESSAGE_FORMAT	Name of notification template (default)
LIMIT_VALUE	Minimum magnitude for a notification to be sent
PRODUCT_TYPE	Type of product to be delivered. If omitted, product is plain text. PDF: PDF from templates
FILENAME	External file to be attached to message
AGGREGATION_GROUP	Notification to be sent based on GROUP type defined by membership in a GROUP. ShakeCast has a predefined CITY group of global cities.
MAX_TRIES	Maximum number of notification attempts.
FACILITY_ID	Unique facility identifier. Text(32)
FACILITY_TYPE	Type of facility. Current defined types are: BRIDGE, CAMPUS, CITY, COUNTY, DAM, DISTRICT, ENGINEERED, INDUSTRIAL, MULTIFAM, ROAD, SINGLEFAM, STRUCTURE, TANK, TUNNEL, UNKNOWN, and HAZUS building types.
EXTERNAL_FACILITY_ID	Organization's unique facility identifier. Text (32) This field must be unique for a facility type but the same external_facility_id may be used for different types of facilities.
FACILITY_NAME	Facility name. Text(128). The value of this field is displayed to the user.
SHORT_NAME	Shortened version of facility name. Text(128). ShakeCast uses the value in this field when a shorter version of the name is needed due to output space limitations.
DESCRIPTION	Facility description. Text(255).
FACILITY_LAT	Facility latitude (decimal degrees, north)
FACILITY_LON	Facility longitude (decimal degrees, east)
GEOM_TYPE	The value of this field is used by ShakeCast to handle the geometry coordinates from the geom field. Text(32) Currently defined types are: POINT, POLYLINE, POLYGON, RECTANGLE, and CIRCLE.
GEOM	Geometry of a facility. The value of this field is used specify the coordinates of the facility. Text(32) Format of this field is in (longitude latitude) pairs separated by a white space. The size limit of data is ~16MB.
BOUND_SOUTH	ShakeMap boundary to south
BOUND_NORTH	ShakeMap boundary to north
BOUND_WEST	ShakeMap boundary to west
BOUND_EAST	ShakeMap boundary to east

Constant	Description
AGGREGATE	Flag to indicate whether notifications should be combined into a single message Integer value e.g., 1
SERVER_ID	Identifier of this ShakeCast server
DNS_ADDRESS	Domain name of this ShakeCast server

Table 17. SYSTEM Notification Keywords

Constant	Description
LOG_MESSAGE_ID	Log sequence ID in ShakeCast database
LOG_MESSAGE_TYPE	Message type in WARNING or ERROR
SERVER_ID	Local ID for this ShakeCast server
DESCRIPTION	Server description
RECEIVE_TIMESTAMP	Timestamp when this notification was requested e.g., yyyy-mm-ddThh:mm:ssZ
NOTIFICATION_ID	Sequence ID in ShakeCast notification table
TRIES	Number of notification attempts.
DELIVERY_STATUS	Result of notification attempt.
SHAKECAST_USER	ShakeCast User ID
DELIVERY_ADDRESS	Email address for delivery
DELIVERY_METHOD	Product delivery type for the given notification EMAIL_HTML: email with html formatting EMAIL_TEXT: plain text email PAGER: simple text message for SMS delivery
NOTIFICATION_TYPE	Types of events that will trigger a notification to be sent Valid notification types: NEW_EVENT: an earthquake exceeding a user-set threshold value. Requires EVENT_TYPE and DELIVERY_METHOD tags. DAMAGE: Triggered when the ground shaking parameter at a facility (or facilities) is between the high and low values of the user-set facility parameters. Requires EVENT_TYPE, DELIVERY_METHOD, and DAMAGE_LEVEL tags. SHAKING: Triggered when the ground shaking parameter at the facility location of the facility exceeds the preset value. Requires EVENT_TYPE, DELIVERY_METHOD, METRIC, and LIMIT_VALUE tags. CAN_EVENT: cancelled event. Requires EVENT_TYPE and

Constant	Description
	<p>DELIVERY_METHOD tags.</p> <p>UPD_EVENT: updated event. Requires EVENT_TYPE and DELIVERY_METHOD tags.</p> <p>NEW_PROD: triggered when a specific ShakeMap product becomes available. Require EVENT_TYPE, DELIVERY_METHOD, and PRODUCT tags.</p>
MESSAGE_FORMAT	Filename of notification template (default)
LIMIT_VALUE	Minimum magnitude for a notification to be sent
AGGREGATE	<p>Flag to indicate whether notifications should be combined into a single message</p> <p>Integer value e.g., 1</p>
AGGREGATION_GROUP	Notification to be sent based on GROUP type defined by membership in a GROUP. ShakeCast has a predefined CITY group of global cities.
MAX_TRIES	Maximum number of notification attempts.
SERVER_ID	Identifier of this ShakeCast server
DNS_ADDRESS	Domain name of this ShakeCast server

Table 18. Derived Value Keywords

Facility Attributes As ATTR_[ATTRIBUTE_NAME]	Description
_ITEMNO	Total number of entries in this notification
NUM[METRIC] (SHAKING/DAMAGE only)	Total number of entries for the specified ShakeMap metric
MIN[METRIC] (SHAKING/DAMAGE only)	The minimum reported value for the specified ShakeMap metric
MAX[METRIC] (SHAKING/DAMAGE only)	The maximum reported value for the specified ShakeMap metric
MEAN[METRIC] (SHAKING/DAMAGE only)	The averaged value for the specified ShakeMap metric
EXCEEDANCE_RATIO (DAMAGE only)	The relative position between the LOW_LIMIT and HIGH_LIMIT values, normalized to between 0 and 1.

